Gate-efficient discrete simulations of continuous-time quantum query algorithms[*]

Dominic W. Berry[†], Richard Cleve[‡], and Sevag Gharibian[‡§]

November 21, 2012

Abstract

We show how to efficiently simulate continuous-time quantum query algorithms that run in time T in a manner that preserves the query complexity (within a polylogarithmic factor) while also incurring a small overhead cost in the total number of gates between queries. By small overhead, we mean T within a factor that is polylogarithmic in terms of T and a cost measure that reflects the cost of computing the driving Hamiltonian. This permits *any* continuous-time quantum algorithm based on an efficiently computable driving Hamiltonian to be converted into a gate-efficient algorithm with similar running time.

1 Introduction and Summary of Result

The continuous-time query model [1] can be thought of as a variant of the standard query model, but where arbitrarily small λ-fractional queries to the data $x_1 x_2 \dots x_L \in \{0,1\}^L$, of the form $|j\rangle \mapsto e^{i\lambda x_j}|j\rangle$, can be made at cost only λ. In the limit as $\lambda \to 0$, such algorithms become continuous-time Hamiltonian evolution processes.

We show that any continuous-time quantum query algorithm whose total query time is T and whose driving Hamiltonian is implementable with G 1- and 2-qubit gates (in a sense defined in Section 3) can be simulated by a discrete-query quantum algorithm using the following resources:

- $O(T \log T / \log\log T)$ queries
- $O(TG\,\log(T) + T\log^3(\|H\|T))$ 1- and 2-qubit gates [or $O(TG\log(T) + TG^3)$ in terms of just T and G]
- $O(\log^3(\|H\|T))$ qubits of space [or $O(G^3)$].

This extends a previous result [2] where the query cost is the same, but where the orders of the second and third resource costs are at least T^2polylog$\,T$ and Tpolylog$\,T$ respectively. The present result can also be compared with the result [3] where the query cost is superior to ours, $O(T)$ (which is asymptotically optimal), but whose methodology does not (as far as we know) yield an efficient gate construction from an efficiently implementable driving Hamiltonian.

[*] Research supported by Canada's NSERC, CIFAR, MITACS, the U.S. ARO, and ARC grant FT100100761.

[†] Department of Physics and Astronomy, Macquarie University, NSW 2109, Australia.

[‡] David R. Cheriton School of Computer Science and Institute for Quantum Computing, University of Waterloo.

[§] Department of Computer Science, University of Illinois at Chicago, USA.

2 Significance to Quantum Computation

The continuous time query model is an important tool for designing algorithms, and for example yielded the algorithm for AND-OR tree evaluation [4]. The difficulty with continuous-time quantum algorithms is that, in order to implement them on quantum computers, these abstract query algorithms need to be translated into concrete algorithms with subroutines substituted for the black-box queries[1]. In these circumstances, what matters is the total gate complexity, which can be large if the cost of the operations performed between the queries is large, even if the number of queries is small. The contribution of our result is that it provides a *systematic* way to obtain a gate-efficient discrete-query algorithm from *any* continuous-time query algorithm where the driving Hamiltonian can be efficiently implemented. That is, whenever the implementation cost of the driving Hamiltonian is small, the total gate complexity is not much more than the query complexity times the cost of implementing each query.

Consider applying the continuous-time quantum algorithm in [4] for AND-OR tree evaluation to evaluate expressions of the form

$$\exists x_1 \forall x_2 \exists x_3 \cdots \forall x_L f(x_1, x_2, \ldots, x_L), \tag{1}$$

where one is given a polynomial (in L) size circuit implementation of $f : \{0,1\}^L \to \{0,1\}$. This corresponds to evaluating a balanced binary AND-OR tree of size $N = 2^L$. A continuous-time query algorithm achieving time $O(\sqrt{N})$ cannot be simulated directly from f, because a small λ-fractional query to f cannot be computed at cost proportional to λ; the algorithm must be efficiently translated into the discrete-query framework to be implementable. But if we substitute the parameters into the simulation in [2], we obtain a gate cost of order $N\text{polylog}\, N$ (losing the square-root speedup) and consume order $\sqrt{N}\text{polylog}\, N$ qubits of space. The simulation in [3] does not appear to yield any bounds less than $O(N)$ on the gate cost. However, our present simulation results in $N^{1/2+o(1)}$ gates and $O(\text{polylog}\, N)$ space (using the fact that the driving Hamiltonian in [4] can be implemented with $N^{o(1)}$ gates). We remark that, for this particular example, a better simulation that is specific to AND-OR tree evaluation (that was discovered after [4]) is known [5].

3 Precise Statement of Main Result

Prior to stating our main result, we give a precise definition of the *implementation cost* of a Hamiltonian acting on l qubits, which is the cost of realising the unitary operation corresponding to evolution under the Hamiltonian from a start time to a finish time. A preliminary idealised definition is as a unitary operation with the following properties. It acts on three registers: a *start time*, a *finish time* and an l-qubit *state*. For any start and finish times t_s and t_f, and any l-qubit state $|\psi\rangle$, the unitary operation maps $|t_s\rangle|t_f\rangle|\psi\rangle$ to $|t_s\rangle|t_f\rangle|\psi'\rangle$, where $|\psi'\rangle$ is the state that results when $|\psi\rangle$ evolves under H from time t_s to time t_f. Assuming that all three registers are finite-dimensional, this can be denoted as a gate as in Fig. 1. We will not require the unitary to be implemented perfectly. We introduce a precision parameter ε', and permit the unitary evolution to be approximated within ε'. This leads to the following definition.

[1] A query is typically *not* something that could be physically implemented directly via continuous-time Hamiltonian evolution, as in an analog quantum computer. A query corresponds to the coherent evaluation of a classical function on several qubits, and requires several quantum gates to implement, regardless of whether it is a full query or a fractional query.

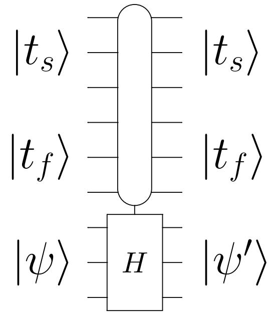

Figure 1: Controlled evolution under Hamiltonian H, with start time t_s, finish time t_f, and target state $|\psi\rangle$.

Definition 3.1. *Let H be a Hamiltonian acting on l qubits. Define H to be* implementable within precision ε' with G gates *if the following unitary operation can be implemented within precision ε' with G elementary gates. The unitary acts on three registers: a start time and finish time, and l qubits set to the initial state. The unitary maps $|t_s\rangle|t_f\rangle|\psi\rangle$ to $|t_s\rangle|t_f\rangle|\psi'\rangle$, where $|\psi'\rangle$ is the state that results when $|\psi\rangle$ evolves under H from time t_s to time t_f. By approximating within ε', we mean with respect to the completely bounded norm.*

We are now ready to state our main result.

Theorem 3.2. (Main) *Let $H(t)$ be a driving Hamiltonian that is approximately implementable within precision $1/T$ using G gates. Then the continuous-time query algorithm can be simulated with constant error by a discrete-query quantum algorithm using the following resources:*

- $O(T \log T / \log \log T)$ *queries*
- $O(TG \log(T) + T \log^3(\|H\|T))$ *1- and 2-qubit gates*
- $O(\log^3(\|H\|T))$ *qubits of space.*

In particular, when G is polylog(T), this is $\tilde{O}(T)$ queries, $\tilde{O}(T)$ 1- and 2-qubit gates, and polylog(T) qubits of space. The norm $\|H\|$ is taken to be $\|H\| := \sup_{t\in[0,T]} \|H(t)\|$ for time-dependent $H(t)$. Because the gate complexity scales linearly in G, we require the driving Hamiltonian to be simulatable efficiently in order for the simulation to be gate-efficient. If, for example, G scaled linearly in $\|H\|$, then the gate complexity would be linear in $\|H\|T$, which is similar to the complexity obtained by product formulae [6]. On the other hand, we have a lower bound of $G = \Omega(\log(\|H\|T))$ (see Section 4.6). As a result, we could express the gate complexity as $O(TG \log(T) + TG^3)$, and the number of qubits of space as $O(G^3)$.

The remaining sections explain our algorithm, with the proof of Theorem 3.2 in Section 6.

4 Compressed CGMSY Construction

We will summarise the construction in [2], and then show how to make it more efficient by compressing the control registers. Before doing so, we state the notation used throughout this paper.

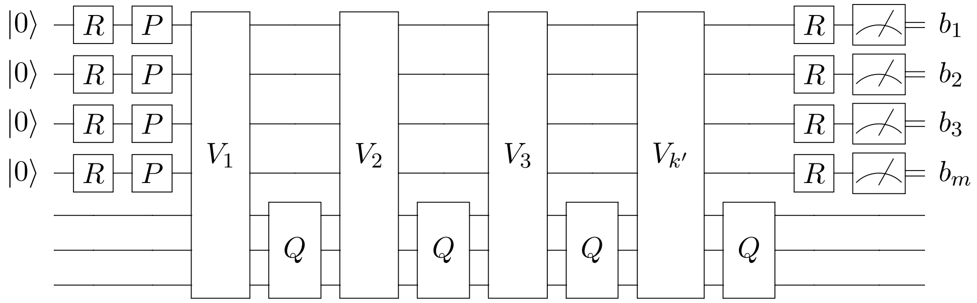

Figure 2: The construction from Ref. [2] to simulate a time interval of length 1/4.

Notation. We denote the sets of linear, Hermitian, and unitary operators acting on complex Euclidean space $\mathcal{X}$ as $\mathcal{L}(\mathcal{X})$, $\mathcal{H}(\mathcal{X})$, and $\mathcal{U}(\mathcal{X})$, respectively. The spectral norm of operator A is $\| A \| := \max\{\| A|v\rangle \|_2 : \| |v\rangle \|_2 = 1\}$. The norm of time dependent operator $A(t)$ is given by $\|A\| = \sup_t \|A(t)\|$. The completely bounded norm, or diamond norm, of superoperator $\Phi : \mathcal{L}(\mathcal{X}) \mapsto \mathcal{L}(\mathcal{Y})$ is defined as $\| \Phi \|_\diamondsuit = \left\| \Phi \otimes I_{\mathcal{L}(\mathcal{X})} \right\|_1$, where the superoperator trace-norm is given by $\| \Phi \|_1 = \max\{\| \Phi(X) \|_{\mathrm{tr}} \mid X \in \mathcal{L}(\mathcal{X}), \| X \|_{\mathrm{tr}} \leq 1\}$. All logarithms are taken to base 2. We define $[m] := \{1, \ldots, m\}$. The tensor product of many zero computational basis states will be represented in compact form as $|0^\ell\rangle := |0\rangle^{\otimes \ell}$.

4.1 Overview of the CGMSY Construction [2]

Our result is obtained by simulating the construction in [2], but by representing some of the qubits in a highly compressed form. This compressed form was known by the authors of [2], but it was not known that all of the steps of the construction can be carried out within the compressed form—especially the *measurement of control qubits*.

The construction in [2] begins with a fractional query algorithm with total query cost T. This is partitioned into segments corresponding to time intervals of the form $[t_0, t_0 + 1/4]$, and with m fractional queries of size $1/4m$ in each such interval. Here, m can be chosen as a power of two without loss of generality; we henceforth assume this is the case. In this work we consider the simulation of each of these segments. In [2] it is shown that each segment can be simulated by a circuit of the form of Fig. 2, where $k' \in O(\log(T)/\log\log(T))$, and whose gates are as follows.

On the first m qubits, collectively called the *control* register,

$$P = \begin{pmatrix} 1 & 0 \\ 0 & i \end{pmatrix}, \qquad R = \begin{pmatrix} \alpha & \beta \\ \beta & -\alpha \end{pmatrix} \qquad \text{with } \beta \approx 1/\sqrt{8m}. \tag{2}$$

The gates labeled Q are the (full) queries; the register they act on is called the *target* register. The gates $V_1, \ldots, V_{k'}$ are the unitaries corresponding to evolving the driving Hamiltonian for various time intervals specified by the control qubits: V_1 for the time interval from t_0 to the position of the first one in the control qubits; V_2 for the time interval delineated by the positions of the first and second ones in the control qubits; and so on. The simulation is successful if $b_1 = \cdots = b_m = 0$, which occurs with probability $\geq 3/4$. The value $\beta^2 = \sin(1/8m) \approx 1/8m$ corresponds to a time interval of 1/4. This time interval is chosen to ensure that the success probability is $\geq 3/4$.

In the case that the simulation is not successful, then there are errors at times corresponding to the b_j that are equal to 1. Reference [2] shows how to correct unsuccessful instances. That analysis continues to hold here without modification. The only subtlety is that we also need to account for the number of gates needed to perform the gates V_j. Each gate may need to be divided into a number of parts corresponding to the number of errors (ones) found. It is shown in Ref. [2] that the average number of ones is $O(1)$, so the *average* number of oracle queries is at most multiplied by a constant factor. Moreover, if the total number of oracle queries permitted is bounded by $O(1/\varepsilon_{\rm tot})$ times the average value, then by the Markov bound, the probability is at least $1 - O(\varepsilon_{\rm tot})$ that the overall correction procedure terminates within this bound [2]. Failure to terminate within the bound can be included in the $\varepsilon_{\rm tot}$ allowable error. For the main result in Theorem 3.2, constant error is considered, so this does not alter the result.

When analysing the complexity due to correcting unsuccessful instances, another factor that needs to be considered is the additional complexity due to correcting the individual errors. The average of this complexity was denoted C_0 in Ref. [2], but an upper bound was not considered. As before, an upper bound equal to $O(1/\varepsilon_{\rm tot})$ times the average value will not be exceeded with probability $1 - O(\varepsilon_{\rm tot})$. Again this does not affect the result in Theorem 3.2 as constant error is considered. As a result of these considerations, when taking into account the corrections, the number of oracle queries and the number of additional 1- and 2-qubit gates are at most multiplied by a constant factor. This means that the correction operations do not alter the scaling, and we do not need to consider them further.

The state of the control registers $R^{\otimes m}|0^m\rangle = (\alpha|0\rangle + \beta|1\rangle)^{\otimes m}$ is highly "compressible" in that most of its amplitude is concentrated on basis states with low Hamming weight. A natural succinct representation of this state is in terms of the positions of the ones in binary. We first define such a succinct form precisely (Section 4.2). We then show how the above circuit can be simulated with the control qubits in their succinct form in these three stages: the initial stage (Section 4.3), which is the construction of the state $R^{\otimes m}|0^m\rangle$; the intermediate stage (Sections 4.4 and 4.5), where $P^{\otimes m}$ is applied to the control qubits and then the queries and driving operations occur; and the final stage (Section 5), which is where the control qubits are measured with respect to the basis $\{R^{\otimes m}|x\rangle : x \in \{0,1\}^m\}$.

4.2 Succinct Representation of Control Qubits

We now propose a succinct encoding scheme which accurately reproduces low Hamming weight basis states. Specifically, consider the set of all m-bit strings whose Hamming weight is at most $k+1$, where k is much smaller than m. The size of this set is bounded above by $(m+1)^{k+1}$. Our encoding scheme utilizes a set of size $(m+1)^{k+1}$ strings to accurately represent this space as follows. We use the notation $|x|$ to denote the Hamming weight of $x \in \{0,1\}^*$. The value of k is chosen to ensure that the error due to omitting high Hamming weight components is no more than ε, and therefore can be taken as

$$k = \Theta\left(\frac{\log(1/\varepsilon)}{\log\log(1/\varepsilon)}\right). \tag{3}$$

We also use a slightly smaller value k' to ensure that the error is no more than ε'; the relation between these primed variables is identical. The Hamming weight cutoff k is used to limit errors that occur repeatedly in the compressed measurement protocol. In contrast, Hamming weight cutoff k' is used to limit errors that only occur once. In particular, we limit the total number of controlled oracle calls to k', because the error due to limiting the Hamming weight there only occurs once. We also limit the number of ones that are measured to k'. The Hamming weight cutoff k is used in our compressed encoding, as the error due to this cutoff will contribute multiple times.

Definition 4.1. *Define the encoding scheme C_q^k on $|x\rangle$ for $x \in \{0,1\}^m$, $|x| \leq k$ as follows. For $x = 0^{s_1}10^{s_2}10^{s_3}\dots 0^{s_h}10^t$, where $h := |x|$, $h \leq k+1$ and $t = m - s_1 - \dots - s_h - h$,*

$$C_m^k|x\rangle = |s_1, s_2, \dots, s_h, \underbrace{m, \dots, m}_{k+1-h}\rangle, \tag{4}$$

where $C_m^k|x\rangle \in (\mathbb{C}^{m+1})^{\otimes k+1}$. For $h > k+1$, $C_m^k|x\rangle$ encodes the positions of the first $k+1$ ones.

4.3 Initialization of Control Qubits in Alternative Encoding

We now show how to simulate the preparation of the state after operation $R^{\otimes m}$ (but before $P^{\otimes m}$, which is deferred to Section 4.5) in succinct form using the encoding of Definition 4.1. To begin, in the original circuit there are m control qubits, whose state is initialised to $(\alpha|0\rangle + \beta|1\rangle)^{\otimes m}$, where $\beta \in \Theta(1/\sqrt{m})$. The amplitudes of terms in this superposition decrease factorially with Hamming weight, and in particular, one can write

$$\begin{aligned}(\alpha|0\rangle + \beta|1\rangle)^{\otimes m} &= \sum_{x\in\{0,1\}^m} \alpha^{m-|x|}\beta^{|x|}|x\rangle \\ &= \sum_{\substack{x\in\{0,1\}^m\\|x|\leq k}} \alpha^{m-|x|}\beta^{|x|}|x\rangle + \sum_{\substack{x\in\{0,1\}^m\\|x|> k}} \alpha^{m-|x|}\beta^{|x|}|x\rangle \\ &= \sum_{\substack{x\in\{0,1\}^m\\|x|\leq k}} \alpha^{m-|x|}\beta^{|x|}|x\rangle + \mu|\nu\rangle, \end{aligned} \tag{5}$$

where, on the last line, $|\nu\rangle$ is orthogonal to every basis state in the sum that precedes it, and $\mu^2 \in 1/2^{O(k)}k!$. Using k as in Eq. (3), $\mu^2 \in O(\varepsilon)$.

Using our encoding scheme C_m^k, we aim to prepare a state of the form

$$\sum_{\substack{x\in\{0,1\}^m\\|x|\leq k}} \alpha^{m-|x|}\beta^{|x|}C_m^k|x\rangle + \mu|\nu'\rangle, \tag{6}$$

for some $|\nu'\rangle \in (\mathbb{C}^{m+1})^{\otimes k+1}$ orthogonal to all the kets arising in the sum. This state has the correct amplitudes for all encodings of strings with Hamming weights up to k. Because we choose k such that $\mu^2 \in O(\varepsilon)$, the error in the encoding is $O(\varepsilon)$.

We now show how to construct an approximation (within distance ε) of the state in Eq. (6) using poly$\,(k, \log m)$ gates. Note that, to accomplish this, we must avoid any approach based on first constructing the expanded state in Eq. (5) then applying C_m^k, since this would immediately entail order m gates. Our efficient approach is to first prepare a state similar to Eq. (5) using a slightly different encoding scheme than C_m^k, denoted B_q^k. We then postprocess the state so that the encoding is changed from B_q^k to C_m^k [i.e. Eq. (6)].

We now introduce the encoding B_q^k by explicit construction. Specifically, it is based on the *exponential superposition* state

$$|\phi_q\rangle := \sum_{s=0}^{q-1} \beta\alpha^s|s\rangle + \alpha^q|q\rangle. \tag{7}$$

The state $|\phi_q\rangle$ is very simple to prepare when $q = 2^r$, as follows. Define the unitary matrix

$$M(\gamma) := \frac{1}{\sqrt{1+\gamma^2}} \begin{pmatrix} 1 & -\gamma \\ \gamma & 1 \end{pmatrix}. \tag{8}$$

Note that

$$\begin{aligned} & M(\alpha^{2^{r-1}}) \otimes \cdots \otimes M(\alpha^2) \otimes M(\alpha)|0^r\rangle \\ & \quad = \frac{\beta}{\sqrt{1-\alpha^{2q}}} \left(|0\ldots00\rangle + \alpha|0\ldots01\rangle + \alpha^2|0\ldots10\rangle + \cdots + \alpha^{q-1}|1\ldots11\rangle\right) \\ & \quad = \frac{1}{\sqrt{1-\alpha^{2q}}} \sum_{s=0}^{q-1} \beta\alpha^s|s\rangle. \end{aligned} \tag{9}$$

Therefore, a circuit that maps $|0^{r+1}\rangle$ to $|\phi_q\rangle$ can be obtained by first applying a one-qubit gate on the first qubit to put it in state $\sqrt{1-\alpha^{2q}}|0\rangle + \alpha^q|1\rangle$, and then applying a sequence of controlled-$M(\alpha^{2^j})$ gates (each controlled by the first qubit being in state $|0\rangle$) to create the state

$$\begin{aligned} & \beta|0\rangle\left(|0\ldots00\rangle + \alpha|0\ldots01\rangle + \alpha^2|0\ldots10\rangle + \cdots + \alpha^{q-1}|1\ldots11\rangle\right) + \alpha^q|1\rangle|0\cdots00\rangle \\ & \quad = \beta\left(|00\ldots00\rangle + \alpha|00\ldots01\rangle + \alpha^2|00\ldots10\rangle + \cdots + \alpha^{q-1}|01\ldots11\rangle\right) + \alpha^q|10\cdots00\rangle \\ & \quad = |\phi_q\rangle. \end{aligned} \tag{10}$$

The reason why state $|\phi_q\rangle$ is useful is because, for $q \geq m$, $|\phi_q\rangle^{\otimes k+1}$ yields a state similar to Eq. (6). The encoding, which we will call $B_q^k|x\rangle$, is slightly different than $C_m^k|x\rangle$, but can be efficiently translated into $C_m^k|x\rangle$ with some "clean-up" operations. Specifically, the encoding is as defined below.

Definition 4.2. *Define the encoding scheme B_q^k on $|x\rangle$ for $x \in \{0,1\}^m$, $|x| \leq k$ as follows. For $x = 0^{s_1}10^{s_2}10^{s_3}\ldots0^{s_h}10^t$, where $h := |x|$, $h \leq k$ and $t = m - s_1 - \cdots - s_h - h$,*

$$B_q^k|x\rangle = |s_1,\ldots,s_h\rangle \left(\sum_{j=0}^{q-t-1} \alpha^j\beta|j+t\rangle + \alpha^{q-t}|q\rangle\right)|\phi_q\rangle^{\otimes k-h}, \tag{11}$$

where $B_q^k|x\rangle \in (\mathbb{C}^{q+1})^{\otimes k+1}$.

The state is then given as in the following theorem.

Theorem 4.3. *For $q \geq m$, the states $B_q^k|x\rangle$ for $|x| \leq k$ are orthonormal and*

$$|\phi_q\rangle^{\otimes k+1} = \sum_{\substack{x\in\{0,1\}^m \\ |x|\leq k}} \alpha^{m-|x|}\beta^{|x|}B_q^k|x\rangle + \mu|\nu'\rangle, \tag{12}$$

for some $|\nu'\rangle$ orthogonal to all $B_q^k|x\rangle$ for $|x| \leq k$.

Proof. The state $|\phi_q\rangle^{\otimes k+1}$ is a superposition of (computational) basis states of the form $|s_1,\ldots,s_{k+1}\rangle$, where $s_1,\ldots,s_{k+1} \in \{0,1,\ldots,q\}$. Intuitively, it is useful to think of each such basis state as an encoding of a binary string $0^{s_1}10^{s_2}1\cdots0^{s_{k+1}}1$ (whose Hamming weight is $k+1$ and length is

$s_1 + \cdots + s_{k+1} + k + 1$). We will show that these basis states can be naturally partitioned into equivalence classes: one for each prefix $x \in \{0,1\}^m$ with $|x| \leq k$, and one for all the remaining basis states.

Let $x \in \{0,1\}^m$ with $h = |x| \leq k$ be of the form $x = 0^{s_1}10^{s_2}10^{s_3}\ldots 0^{s_h}10^t$. Consider the set P_x that consists of all $|s'_1, s'_2, \ldots, s'_{k+1}\rangle$ that are encodings of strings whose m-bit prefix is x. The set P_x consists of all $|s'_1, s'_2, \ldots, s'_{k+1}\rangle$ such that $(s'_1, s'_2, \ldots, s'_h) = (s_1, s_2, \ldots, s_h)$, $s'_{h+1} \in \{t, \ldots, q\}$, and $s'_{h+2}, \ldots, s'_{k+1} \in \{0, \ldots, q\}$. It follows that the sum of all the terms in the superposition

$$|\phi_q\rangle^{\otimes k+1} = \sum_{s'_1=0}^{q} \sum_{s'_2=0}^{q} \cdots \sum_{s'_{k+1}=0}^{q} \alpha^{s'_1+s'_1+\cdots+s'_{k+1}} \beta^{|\{\ell | s'_\ell < q\}|} |s'_1, s'_2, \ldots, s'_{k+1}\rangle \tag{13}$$

that correspond to elements of P_x is

$$\begin{aligned} &\alpha^{s_1}\beta \cdots \alpha^{s_h}\beta |s_1, \ldots, s_h\rangle \left(\sum_{j=t}^{q-1} \alpha^j \beta |j\rangle + \alpha^q |q\rangle \right) |\phi_q\rangle^{\otimes k-h} \\ &\quad = \alpha^{s_1}\beta \cdots \alpha^{s_h}\beta\alpha^t |s_1, \ldots, s_h\rangle \left(\sum_{j=0}^{q-t-1} \alpha^j \beta |j+t\rangle + \alpha^{q-t} |q\rangle \right) |\phi_q\rangle^{\otimes k-h} \\ &\quad = \alpha^{m-|x|}\beta^{|x|} B_q^k |x\rangle, \end{aligned} \tag{14}$$

which is the appropriate weighting for $B_q^k|x\rangle$ in the sum in Eq. (12).

Thus, the basis states in the superposition in Eq. (13) corresponding to encodings of strings $x \in \{0,1\}^m$ of Hamming weight at most k can be grouped into equivalence classes P_x. What about the remaining terms in $|\phi_q\rangle^{\otimes k+1}$ which do not fall in any P_x? These are the $|s_1, \ldots, s_{k+1}\rangle$ where $s_1 + \cdots + s_{k+1} + k + 1 \leq m$. Therefore, we can set

$$\mu|\nu'\rangle = \sum_{s_1+\cdots+s_{k+1}+k+1 \leq m} \alpha^{s_1+\cdots+s_{k+1}} \beta^{k+1} |s_1, \ldots, s_{k+1}\rangle, \tag{15}$$

where $\mu \in \mathbb{R}$ is chosen so that $|\nu'\rangle$ is normalised. All the $B_q^k|x\rangle$ and $|\nu'\rangle$ are mutually orthogonal since they are constructed from a partition of the basis states. □

4.4 Converting from the B Encoding to the C Encoding

We have thus far shown how to prepare states in the encoding B_q^k. As mentioned above, we can now convert from the encoding B_q^k to our desired encoding C_m^k. This is achieved by "cleaning up" the registers that follow register $h = |x|$ in $B_q^k|x\rangle$ [compare Eq. (4) with Eq. (11)]. The difference is that, instead of these registers being in the state $|m\rangle$, they are in the state $|\phi_q\rangle$ (for registers $h+2$ to $k+1$). Register $h+1$ is in a state that is similar to $|\phi_{q-t}\rangle$, except that the basis states are shifted by t. Therefore, we need a way of converting these registers to the state $|m\rangle$. However, this conversion depends on both h and t, so we first need these quantities.

We will first give a simplified explanation, then expand on the technical details. To determine h and t, we compute the prefix sums

$$|s_1\rangle|s_2\rangle \cdots |s_{k+1}\rangle \mapsto |s_1 + 1\rangle|s_1 + s_2 + 2\rangle \cdots |s_1 + s_2 + \cdots + s_{k+1} + k + 1\rangle. \tag{16}$$

This gives the *absolute* positions of the ones. The value of h can be determined by finding the first register with a value larger than m (which would give a position for a one past the end of the string).

Now we can identify register $h+1$. For this register, we wish to subtract t, so that the state of this register [as in Eq. (11)] becomes $|\phi_{q-t}\rangle$. At this stage we have computed the prefix sums, and subtracting $m+1$ from this modified register gives the same result as subtracting t from the unmodified register. That is, we do not need to explicitly compute t to subtract it, because it is obtained implicitly in the prefix sum. For all the other registers we then undo the prefix sums.

At this stage we have h in an ancilla, and we have subtracted t from register $h+1$. Now we can undo the procedure to prepare $|\phi_q\rangle$ in registers $h+1$ to $k+1$. Register $h+1$ is actually in state $|\phi_{q-t}\rangle$ rather than $|\phi_q\rangle$, but it is a good approximation of state $|\phi_q\rangle$. Therefore the inverse preparation yields states $|0\rangle$ in registers $h+1$ to $k+1$, with this being approximate for register $h+1$. It is trivial to convert $|0\rangle$ to $|m\rangle$, then uncompute the value of h in the ancilla register. This then completes the conversion of the encoding.

In summary the overall procedure is as follows.

1. Compute the prefix sums.
2. Compute $h = |x|$ in an ancilla register.
3. Uncompute the prefix sums for registers other than $h+1$, and subtract $m+1$ from register $h+1$.
4. Invert the procedure to prepare $|\phi_q\rangle$ from $|0\rangle$ on registers $h+1$ to $k+1$, and swap register $h+1$ with the error flag register.
5. Flip one qubit on registers $h+1$ to $k+1$ to change $|0\rangle$ to $|m\rangle$.
6. Uncompute h in the ancilla register.

Next we explain the technical details, including the error flag register. When computing the prefix sums, we can first consider the case of low-Hamming weight strings with $h \le k$. For the first h registers the result is at most m, whereas for register $h+1$, the result is (coherently) more than m. To prevent the value in register $h+1$ wrapping around modulo m, we instead expand the registers to dimension $m+q+2$, and perform the computations modulo $m+q+2$. Because the value in register $h+1$ is no more than that in h (which is $\le m$) plus $q+1$, the value is $\le m+q+1$, and does not wrap around modulo $m+q+2$. The values in registers $h+2$ to $k+1$ may wrap around, but this does not affect the calculation. This covers steps 1 and 2 above.

Next, considering step 3, the value in register $h+1$ will be

$$s_1 + \ldots + s_h + s_{h+1} + h + 1 = m - t + s_{h+1} + 1. \tag{17}$$

We aim to obtain $s_{h+1} - t$ in this register. If we had computed the value of t, we could uncompute the prefix sums, then subtract t. However, it is obvious from Eq. (17) that we can just subtract $m+1$ instead. Note that this is the first register that is larger than m, so subtracting $m+1$ does not result in a negative number. We also need to uncompute the prefix sums for the other registers. This can be achieved by working backwards from register $k+1$ to $h+2$ uncomputing prefix sums, subtracting $m+1$ from register $h+1$, then uncomputing prefix sums from register h back to 1.

Next we consider the inverse preparation in step 4. At this stage, we have subtracted t from register $h+1$ yielding the exponential state

$$|\phi_{q-t}\rangle = \sum_{s=0}^{q-t-1} \beta\alpha^s |s\rangle + \alpha^{q-t}|q-t\rangle. \tag{18}$$

By choosing q to be sufficiently large, $|\phi_{q-t}\rangle$ is close to $|\phi_q\rangle$, and inverting the procedure for preparing $|\phi_q\rangle$ yields an accurate approximation of $|0^{r+1}\rangle$. To be more precise, note that $\langle\phi_{q-t}|\phi_q\rangle = 1-(1-\beta)\alpha^{2(q-t)}$. Therefore, we have $\langle\phi_{q-t}|\phi_q\rangle \geq 1-\varepsilon$ if $q \geq m+(1/\beta^2)\log(1/\varepsilon)$. To achieve this, $|\phi_q\rangle$ need only consist of $\log(m+1/\beta^2)+\log\log(1/\varepsilon)+O(1)$ qubits. In particular, in our context where $\beta = \Theta(1/\sqrt{m})$, the number of qubits is $\log m + \log\log(1/\varepsilon)+O(1)$, so the precision scales double exponentially with the number of additional qubits beyond $\log m$.

This approximate step could alternatively be performed using the state preparation procedure of Grover and Rudolph [7]. Another alternative is to use amplitude amplification to ensure that the register is set to zero correctly. These alternatives would also not be exact, because they would require the coherent calculation of trigonometric functions.

It is convenient for the analysis to swap register $h+1$ with an "error flag" register that has been prepared in the $|0\rangle$ state. Then, if this register is measured as not zero, it flags that the clean-up operation has not occurred properly. On the other hand, register $h+1$ *is* exactly $|0\rangle$.

We also need to take account of the action of the conversion procedure on the state $|\nu'\rangle$. This state is a superposition of basis states $|s_1,\ldots,s_{k+1}\rangle$, where $s_1+\ldots+s_{k+1}+k+1 \leq m$. This means that, when we compute the prefix sums, the last register will *not* be $> m$. In this case, we can set $h = k+1$, and then make no changes to the other registers in steps 3 to 5 for this value of h. This means that $|\nu'\rangle$ is unchanged. The exact form of this state is unimportant, because it corresponds to an error. However, $|\nu'\rangle$ is a superposition of strings of Hamming weight $h+1$ encoded using C_m^k, and remains so under the conversion.

In summary, the overall preparation procedure is to prepare the state $|\phi_q\rangle^{\otimes k+1}$, then perform the clean-up procedure consisting of steps 1 to 6 above. By choosing $\log q \in \Theta(\log m + \log\log(1/\varepsilon))$ (for q a power of two), this then yields the state (6) within distance $O(\varepsilon)$. Our circuit has size

$$O\left(k\left[\log m + \log\log(1/\varepsilon)\right]\right). \tag{19}$$

The final state has no values in its registers larger than m, so it can be stored in registers of dimension $m+1$, though higher dimensions are required in intermediate steps.

To prepare the state, we have started with all qubits of registers in the state $|0\rangle$. It is convenient to start with these registers in the state $|m\rangle$, flip one qubit in each register to give $|0\rangle$, then perform the preparation procedure as described above. Then we are mapping the state $C_m^k|0^m\rangle$ (which is the state $|m\rangle^{\otimes k+1}$) to the succinct representation of $(\alpha|0\rangle+\beta|1\rangle)^{\otimes m}$ as defined in Eq. (6).

4.5 Phase Gates, Queries and Driving Operations

Applying the phase gates, $P^{\otimes m}$, to the control qubits in their succinct representation is straightforward because $P^{\otimes m}|x\rangle = i^{|x|}|x\rangle$. We need only compute $|x|$ in an ancilla register, apply $|s\rangle \mapsto i^s|s\rangle$, and then uncompute $|x|$ in the ancilla.

To apply the driving operations, we note that our definition of driving Hamiltonian implementation fits perfectly in this context, once we compute the prefix sums to give the positions of the ones, as in Eq. (16). In the compressed representation, V_1 is the implementation of the driving Hamiltonian with t_s hardwired to 0 and t_f controlled by the first register. V_2 is the implementation with t_s controlled by the first register and t_f controlled by the second register, and so on. At the end, the prefix sums can be uncomputed.

4.6 The Value of m Needed

In the CGMSY construction the number of fractional queries m comes from breaking up the evolution under the oracle and the driving Hamiltonian via a product formula. To obtain error bound by $\varepsilon_{\rm tot}$ with evolution over time T and driving Hamiltonian with norm $\|H\|$, the number of time intervals needed in a Trotter-Suzuki product formula for constant Hamiltonian H is $O(\|H\|T(\|H\|T/\varepsilon_{\rm tot})^{\delta})$ [6]. For the CGMSY construction the intervals need to be of equal size, which restricts δ to 1/2.

For time-dependent Hamiltonians, the complexity of Trotter-Suzuki product formulae will depend on the magnitude of the derivatives of H when one is sampling the Hamiltonians at different times [8]. The situation we have here is somewhat different, because we assume that the evolution under the time-dependent driving Hamiltonian can be implemented. In this case, the error does not depend on the time derivative, and the error for a short time interval δt can be bounded as $\|H\|\delta t^2$ (this is easily derived from Eq. (2.3) of Ref. [9]). Hence the number of intervals to limit the overall error to $O(\varepsilon_{\rm tot})$ need be no greater than $O(\|H\|T^2/\varepsilon_{\rm tot})$. The number of intervals in one CGMSY segment of length $O(1)$ is therefore $m = O(\|H\|T/\varepsilon_{\rm tot})$.

Another question is the precision that the time needs to be specified to in order to limit the overall error to $\varepsilon_{\rm tot}$. It is easily shown that the error in the time needs to be bound as $O(\varepsilon'/\|H\|)$ in order to limit the error in a single operation to ε'. If the time is being specified on the interval $[0, T]$, then the number of bits needed for the time is $\lceil \log(\|H\|T/\varepsilon')\rceil$. Because there are $O(1)$ controlled Hamiltonian evolutions in each CGMSY segment, we need $\varepsilon' = O(\varepsilon_{\rm tot}/T)$. This gives the number of bits for the time as $\log(\|H\|T^2/\varepsilon_{\rm tot}) + O(1)$ (where the constant $O(1)$ is because ε' may have a constant of proportionality with $\varepsilon_{\rm tot}/T$).

This result is consistent with the value of m used, because $\log(\|H\|T^2/\varepsilon_{\rm tot}) + O(1)$ bits are needed to specify an integer from 0 to $O(mT)$. In the CGMSY construction, a superposition over the m time intervals is used, so the number of qubits needed is $\lceil \log m\rceil$. The number of the CGMSY segment also needs to be stored, but that can be stored in $O(\log T)$ classical bits.

One can use the number of bits for the time to place a lower bound on the complexity of implementing the driving Hamiltonian. To obtain overall accuracy $O(\varepsilon_{\rm tot})$, the driving Hamiltonian needs accuracy of $O(\varepsilon_{\rm tot}/\|H\|T)$ in the time. There are $\Theta(\|H\|T^2/\varepsilon_{\rm tot})$ starting and finishing times, so by a counting argument the gate complexity is $\Omega(\log(\|H\|T/\varepsilon_{\rm tot}))$. If the driving Hamiltonian is constant, then it is only the length of the time which is important, and that is limited to $O(1)$. The number of times is then $\Theta(\|H\|T/\varepsilon_{\rm tot})$, but the lower bound on the complexity is still $\Omega(\log(\|H\|T/\varepsilon_{\rm tot}))$. For constant error we therefore have $G = \Omega(\log(\|H\|T))$, as used in Section 3.

5 Measurement of the Control Qubits

What remains is to perform the final measurement. This should logically correspond to what happens if the state is decoded from its succinct representation to m qubits and then, for each qubit, an R gate is applied and it is measured in the computational basis. Of course, this cannot be literally implemented this way, because it would increase the gate and space usage to at least m; our task is to *logically* perform this while remaining in the succinct representation.

Recall now that in Section 4.3, we constructed a procedure that approximately prepares $R^{\otimes m}|0^m\rangle$ in succinct form [see Eq. (6)]. We define U_m to be the ideal unitary that would exactly prepare the state (6). The action of the ideal state preparation procedure is then $U_m C_m^k|0^m\rangle \approx C_m^k R^{\otimes m}|0^m\rangle$. The procedure we have described does not exactly perform this unitary, but it is within distance $O(\varepsilon)$. Also, we do not have an exact equality, because representations of terms with Hamming weight greater than k in $R^{\otimes m}|0^m\rangle$ are not obtained with the correct weights. More precisely, we have

$$U_m C_m^k|0^m\rangle = \sum_{\substack{x\in\{0,1\}^m \\ |x|\leq k}} \alpha^{m-|x|}\beta^{|x|} C_m^k|x\rangle + \mu|\nu'\rangle. \tag{20}$$

This is to be compared with the uncompressed setting [Eq. (5)], in which we have

$$R^{\otimes m}|0^m\rangle = \sum_{\substack{x\in\{0,1\}^m \\ |x|\leq k}} \alpha^{m-|x|}\beta^{|x|}|x\rangle + \mu|\nu\rangle. \tag{21}$$

In terms of the *logical* data, U_m and $R^{\otimes m}$ produce almost the same state when applied to $|0^m\rangle$.

Returning to the issue of measurement, in the *uncompressed* basis we would like to perform $R^{\otimes m}$, then perform a computational basis measurement. In the particular case that the computational basis measurement yielded all zeros, the measurement operator is $|0^m\rangle\langle 0^m|R^{\otimes m}$. Because we are performing all operations in the compressed basis, this measurement operator can be represented by $C_m^k|0^m\rangle\langle 0^m|R^{\otimes m}(C_m^k)^\dagger$. Because R is self-inverse, this is approximately the same as $C_m^k|0^m\rangle\langle 0^m|(C_m^k)^\dagger U_m^\dagger$. That is, to achieve this measurement result we first invert the preparation procedure described by U_m. Then, because $C_m^k|0^m\rangle = |m\rangle^{\otimes k+1}$ is a computational basis state, we can achieve the desired result by performing a computational basis measurement.

Ideally, this is what we want, but we also need to be able to find the positions of the ones in the case that the all-zero string is not obtained. At first glance, one might imagine that applying $U_m^\dagger$ in place of $R^{\otimes m}$ would yield a succinct representation of the final outcome state, so measuring in the computational basis would provide the correct result. Unfortunately, this does not accurately simulate the final measurement except in the case where the all-zero string is obtained. The problem is that U_m and $R^{\otimes m}$ are *only* in close agreement when applied to the logical state $|0^m\rangle$. For any other logical state $|x\rangle$ (for non-zero $x \in \{0,1\}^m$), applying U_m and $R^{\otimes m}$ need not yield states in any close agreement.

Our first observation towards overcoming this problem is that we can at least perform an *incomplete* measurement that captures a seemingly small part of what we are seeking: we can cause the state to either collapse to logical $|0^m\rangle$ or to the subspace that is the orthogonal complement of this state—and with the correct probabilities. This is achieved by performing $U_m^\dagger$ and then the 2-outcome incomplete projective measurement that distinguishes between the logical state $|0^m\rangle$ and its orthogonal complement $|0^m\rangle^\perp$, and then applying U_m to the resulting collapsed state. Our method to complete the measurement is to apply the above procedure recursively, on the two halves of the logical string. We now first motivate this procedure intuitively, followed by further technical details and a rigorous proof of correctness.

5.1 Measuring in Succinct Form: Intuition

The intuition behind our measurement strategy is given by the following simple thought experiment. Consider the problem of measuring an m-qubit state $|\psi\rangle$ in the computational basis. This can be accomplished by performing a sequence of two-outcome measurements in a variety of ways. One obvious approach is to measure the state of the first qubit, then the second qubit, and so on. Each final outcome $x \in \{0,1\}^m$ will occur with exactly the same probability as with the original complete measurement. We now describe an alternative—and unconventional—approach for simulating the same measurement.

First, perform the measurement distinguishing between $|0^m\rangle$ and $|0^m\rangle^\perp$, its orthogonal complement. If the state collapses to $|0^m\rangle$ we halt, outputting 0^m. Otherwise (when the state collapses to $|0^m\rangle^\perp$), apply the measurement $|0^{m/2}\rangle$ vs. $|0^{m/2}\rangle^\perp$ to the first $m/2$ qubits. If that part of the state collapses to $|0^{m/2}\rangle$ then output $0^{m/2}$ for the first $m/2$ bits; otherwise recurse further. Once this recursive measurement procedure for the first $m/2$ qubits has terminated, repeat it for the second $m/2$ qubits. Each final outcome $x \in \{0,1\}^m$ occurs with exactly the same probability as with the original complete measurement. Note that although this process may appear complicated, it terminates fast whenever the Hamming weight of the final outcome x is small: for Hamming weight up to k', at most $k' \log m$ steps are performed.

Our actual scenario is different than the one described above in that the final measurement is in the basis $\{R^{\otimes m}|x\rangle : m \in \{0,1\}^m\}$ rather than the computational basis. However, our logical U_m and $U_m^\dagger$ permit us to approximate the $R^{\otimes m}|0^m\rangle$ vs. $R^{\otimes m}|0^m\rangle^\perp$ measurement well. Also, making use of the fact that the underlying operation that we are simulating has a tensor product structure, $R^{\otimes m}|x_1 x_2\rangle = R^{\otimes m/2}|x_1\rangle R^{\otimes m/2}|x_2\rangle$ for any $x_1, x_2 \in \{0,1\}^{m/2}$, we can emulate the recursive procedure in the above thought experiment. We now make this rigorous.

5.2 Measuring in Succinct Form: Details

We now introduce Alg. 5.2, which formalises the intuition behind the recursive measurement outlined above, and show that it simulates the desired measurement in succinct form. Recall that we assume without loss of generality that m is a power of 2.

Before stating Alg. 5.2, we require a lemma which allows us to efficiently "split" the encoded version of string $x = x_1 x_2$ into the concatenation of the encoded versions of x_1 and x_2.

Lemma 5.1. *Let $x = x_1 x_2$ for $x \in \{0,1\}^m$ with $|x| \leq k$, $x_1, x_2 \in \{0,1\}^{m/2}$ and m a power of 2. Then there exists a quantum circuit with complexity $O(k \log m)$ for achieving the mapping*

$$C_m^k |x_1 x_2\rangle \mapsto C_{m/2}^k |x_1\rangle \otimes C_{m/2}^k |x_2\rangle, \tag{22}$$

where $C_m^k|x_1 x_2\rangle$, $C_{m/2}^k|x_1\rangle$, $C_{m/2}^k|x_2\rangle \in (\mathbb{C}^{m+1})^{\otimes k+1}$.

Proof. Because both $C_m^k|x_1 x_2\rangle$ and $C_{m/2}^k|x_1\rangle \otimes C_{m/2}^k|x_2\rangle$ are computational basis states, the procedure that is performed is the same as would be performed classically, except that it must be performed coherently. That is, there is a reversible classical procedure to split the encoding in the computational basis, which immediately provides a coherent procedure for splitting the encoding. Because there are $O(k)$ registers of size $O(\log m)$, the complexity of this procedure is $O(k \log m)$. □

The formal statement of the recursive measurement algorithm is given in Alg. 5.2. To perform our recursive measurement, we simply call MEASURE$(A, 1, m)$, where A is the register containing

Algorithm 5.2. $S = \text{MEASURE}(\ A\ ,\ m_1\ ,\ m_2\ \ )$.

- Input: A – Registers corresponding to space $(\mathbb{C}^{m+1})^{\otimes k+1}$ containing the subset $\{m_1, \ldots, m_2\}$ of the encoded control qubits.
  m_1 – The starting index $m_1 \in [m]$ of the encoded qubits in A.
  m_2 – The ending index $m_2 \in [m]$ of the encoded qubits in A.
- Precondition: $m_2 - m_1 + 1$ is a power of two.
- Output: A set of indices $S \subseteq [m]$ containing the positions where an uncompressed measurement would have found ones in the uncompressed setting.

Perform a measurement described by the measurement operators $M_{\text{c},0}^{m_2-m_1+1}$ and $M_{\text{c},1}^{m_2-m_1+1}$, where $M_{\text{c},0}^{n} := U_n C_n^k |0^n\rangle\langle 0^n|(C_n^k)^\dagger U_n^\dagger$ and $M_{\text{c},1}^{n} := I - M_{\text{c},0}^{n}$. Label the measurement result d. Then

1. (Zero detected) If $d = 0$: Return $S = \emptyset$.
2. (Base case) If $d = 1$ and $m_1 = m_2$: Return $S = \{m_1\}$.
3. (Recurse) If $d = 1$ and $m_2 > m_1$: Split A to A_1 and A_2, containing the encoded forms of the first and second halves, respectively, of the control qubits. Then return

$$S = \text{MEASURE}(A_1, m_1, (m_1 + m_2 - 1)/2) \cup \text{MEASURE}(A_2, (m_1 + m_2 + 1)/2, m_2). \quad (23)$$

our compressed control qubits. Once the procedure finishes running, it will return the locations of all the ones an uncompressed measurement would have obtained when measuring the uncompressed version of A. We truncate the recursive measurement procedure if k' ones have been located, to limit the complexity of the procedure.

We now introduce a notation that will be used throughout the remainder of the paper in order to simplify reference to quantities in the uncompressed protocol versus the compressed protocol. For quantities (states, operators or probabilities) in the compressed protocol, we will use a superscript or subscript "c", whereas we will use "u" for the uncompressed protocol. To refer to quantities defined for both, we will use "η". We also use n to refer to operations acting on a compressed sub-portion of the string of length n (instead of m for the full string).

To perform the measurement described by the measurement operators $M_{\text{c},d}^{n}$ in Alg. 5.2, we apply $U_n^\dagger$, perform the measurement that distinguishes the encoded all-zero state from all other states, then apply U_n. In this form it is clear why we need to perform the operation U_n after the measurement: it means that all states orthogonal to that corresponding to measurement result 0 are unchanged, because they are just acted upon by the identity. The final U_n operation is also included for the 0 measurement result for simplicity, but it is not needed. As these measurement operators are projections, they are the same as the positive operator-valued measure elements.

For simplicity we have described the measurement in terms of U_n and $U_n^\dagger$, but in reality we will use operations on an expanded space that includes an error-flag ancilla. Recall that, because $|\phi_{q-t}\rangle$ is not exactly equal to $|\phi_q\rangle$, we have a register that is not exactly reset to zero, and this is swapped into an ancilla register. The unitary operations in this expanded space will be denoted $\widetilde{U}_n$ and $\widetilde{U}_n^\dagger$.

Then the action of $\widetilde{U}_n$ is

$$\widetilde{U}_n C_n^k|0\rangle \otimes |0\rangle = \sum_{\substack{x\in\{0,1\}^n \\ |x|\le k+1}} \xi_x^n \left[\sqrt{1-\varepsilon_x} C_n^k|x\rangle \otimes |0\rangle + \sqrt{\varepsilon_x}|\mathrm{err}_x\rangle \otimes |1\rangle\right]. \tag{24}$$

Here the tensor product with $|0\rangle$ on the left-hand side indicates the use of ancillas that are initially in the state zero. The amplitudes ξ_x^n are the amplitudes for each $C_n^k|x\rangle$ in the state (6) (when m is replaced with n). These amplitudes include those for $|x| = k+1$ for the state $|\nu'\rangle$, which corresponds to encoded Hamming-weight $k+1$ states. For $|x| \le k$, we have $\xi_x^n = \alpha^{n-|x|}\beta^{|x|}$. The tensor product with $|0\rangle$ on the right-hand side indicates ancillas that will be set to zero in the case of success. The parameter ε_x is $\le \varepsilon$, and can in general depend on x. The state $|\mathrm{err}_x\rangle$ is an error state.

For the ideal state preparation, we have

$$\langle x|(C_n^k)^\dagger U_n C_n^k|0\rangle = \xi_x^n. \tag{25}$$

Using the expression for the action of $\widetilde{U}_n$, we find

$$[(\langle x|(C_n^k)^\dagger) \otimes \langle 0|][\widetilde{U}_n C_n^k|0\rangle \otimes |0\rangle] = \xi_x^n\sqrt{1-\varepsilon_x} = \langle x|(C_n^k)^\dagger U_n C_n^k|0\rangle[1-O(\varepsilon)]. \tag{26}$$

There is no contribution from the error register, because the error flag is orthogonal to zero for that register.

To perform the measurement, we append ancillas in the zero state, and perform $\widetilde{U}_n^\dagger$. Then we perform the measurement that projects onto $C_n^k|0^n\rangle \otimes |0\rangle$ and its orthogonal complement. Here the tensor product with $|0\rangle$ indicates the extra ancillas used by the full preparation procedure $\widetilde{U}_n$. Then we perform $\widetilde{U}_n$.

The action of this measurement will have error $O(\varepsilon)$ from that used in the algorithm. First, consider the resulting state for zero measurement result and initial state $C_n^k|x\rangle$.

$$\begin{aligned}
\widetilde{U}_n[C_n^k|0^n\rangle\langle 0^n|(C_n^k)^\dagger \otimes |0\rangle\langle 0|]\widetilde{U}_n^\dagger C_n^k|x\rangle|0\rangle &= \widetilde{U}_n C_n^k|0^n\rangle \otimes |0\rangle[\langle x|\langle 0|(C_n^k)^\dagger \widetilde{U}_n C_n^k|0^n\rangle|0\rangle]^* \\
&= \widetilde{U}_n C_n^k|0^n\rangle \otimes |0\rangle[\langle x|(C_n^k)^\dagger U_n C_n^k|0^n\rangle]^*[1-O(\varepsilon)] \\
&= \widetilde{U}_n C_n^k|0^n\rangle \otimes |0\rangle[\langle 0^n|(C_n^k)^\dagger U_n^\dagger C_n^k|x\rangle][1-O(\varepsilon)].
\end{aligned} \tag{27}$$

Therefore we find that the probability of this result is changed by no more than $O(\varepsilon)$. In addition, tracing over the ancillas used, $\widetilde{U}_n C_n^k|0^n\rangle \otimes |0\rangle$ is an approximation of $U_n C_n^k|0^n\rangle$ with trace distance $O(\varepsilon)$. Therefore, for the zero measurement result, the resulting state has trace distance no more than $O(\varepsilon)$ from that for the ideal measurement using U_n.

The resulting state for measurement result 1 is then

$$C_n^k|x\rangle|0\rangle - \widetilde{U}_n C_n^k|0^n\rangle \otimes |0\rangle[\langle 0|(C_n^k)^\dagger U_n^\dagger C_n^k|x\rangle][1-O(\varepsilon)]. \tag{28}$$

This is because $\widetilde{U}_n$ is exactly the inverse of $\widetilde{U}_n^\dagger$ in the expanded space. Trivially from the result for the zero measurement result, once we trace over the ancilla the resulting state has trace distance no more than $O(\varepsilon)$ from that for the ideal measurement. As a result, even though we can not perform U_n exactly, we can approximate the measurements within error ε using the $\widetilde{U}_n$ and $\widetilde{U}_n^\dagger$ operations.

To show that the algorithm correctly simulates the desired uncompressed measurement, we consider a similar recursive measurement on the uncompressed state. We show that, except for the imprecision due to approximating U_n and omitting high Hamming weight components, the low Hamming weight portions of the states in Eqs. (20) and (21) evolve identically. Moreover, this holds even if the control qubits are entangled with a target register, as is generally the case here.

In the uncompressed setting, the state of the control and target registers before the final measurement can be described as approximately

$$|\widetilde{\psi}_{\mathrm{u}}\rangle := \sum_{\substack{x\in\{0,1\}^m \\ |x|\leq k}} \gamma_{x,0}^{()}|x\rangle|w_x\rangle, \tag{29}$$

where $|w_x\rangle$ describes the state of the target register where the queries Q are applied. Note that $|\widetilde{\psi}_{\mathrm{u}}\rangle$ is unnormalised, as we have omitted the high Hamming weight component. Similarly, in the compressed setting, before the final measurement we approximately have the (unnormalised) state

$$|\widetilde{\psi}_{\mathrm{c}}\rangle := \sum_{\substack{x\in\{0,1\}^m \\ |x|\leq k}} \gamma_{x,0}^{()}C_m^k|x\rangle|w_x\rangle, \tag{30}$$

where the states $|w_x\rangle$ coincide with those in the uncompressed case. The coefficients $\gamma_{x,0}^{()}$ are the same in each case, and are equal to $i^{|x|}\xi_x^m$. We use this notation for consistency with the coefficients for the intermediate states in Eqs. (32) and (33) below.

We consider a measurement in the uncompressed case that is the same as in Alg. 5.2. We show that the results obtained in the two cases are close, but there are two sources of error: (1) the error incurred due to the high Hamming weight component of the state, and (2) the error due to not implementing U_n exactly. First we discuss the *error-free* case, i.e. where (1) we omit the high Hamming weight component, and where (2) U_n is implemented exactly. We subsequently reintroduce both sources of error and analyse their impacts. In the error-free analysis, we show the following.

Theorem 5.3 (Error-free simulation)**.** *Assume we are in the error-free setting defined above. Then, suppose that before the final measurement, the states of the uncompressed and compressed control and target qubits are given by Eqs. (29) and (30), respectively. Then, Alg. 5.2 exactly simulates the uncompressed $R^{\otimes m}$ measurement in the following sense:*

1. *After running Alg. 5.2, the probability of obtaining a given measurement result is the same as for the uncompressed $R^{\otimes m}$ measurement, and*

2. *for a given measurement result the state of the target register in both uncompressed and compressed settings matches.*

Proof. Measuring $R^{\otimes m}|\widetilde{\psi}\rangle$ in the computational basis can also be simulated using a recursive approach; namely, we apply $R^{\otimes m}$, followed by the incomplete measurement of $|0^m\rangle$ versus its orthogonal complement, then apply $R^{\otimes m}$. This can be represented by the measurement operators $M_{\mathrm{u},d}^m$, with

$$M_{\mathrm{u},0}^n := R^{\otimes n}|0^n\rangle\langle 0^n|R^{\otimes n}, \tag{31}$$

and $M_{\mathrm{u},1}^n := I - M_{\mathrm{u},0}^n$. Similar to Alg. 5.2, we are including the application of $R^{\otimes m}$ for both measurement results for simplicity, though it is not needed for result 0. If we obtain 1 as the outcome, we recurse on the two blocks of $m/2$ qubits by applying the measurement with operators $M_{\mathrm{u},d}^{m/2}$, and so forth.

To prove the result, we simply need to show that at each step in the recursion the states resulting from measurement operators $M^n_{\mathrm{c},d}$ and $M^n_{\mathrm{u},d}$ are equivalent. Let us denote the measurement result obtained at each step in the recursive measurement scheme by d_j. Then, at step ℓ, we have measurement results $d_1, \ldots, d_{\ell-1}$, and will have a state that depends on those measurement results. Let us assume that at this step we have equivalent states for the compressed and uncompressed cases. The base case is that for $\ell = 1$, where the initial states (29) and (30) are equivalent. Then the states for the two cases can be expressed as

$$|\psi^{(d_1,\ldots,d_{\ell-1})}_{\mathrm{c},\ell-1}\rangle = \sum_{|x|\le k} \gamma^{(d_1,\ldots,d_{\ell-1})}_{x,\ell-1} (C^k_n \otimes C^k_{\mathrm{rest}})|x\rangle|w_x\rangle, \tag{32}$$

$$|\psi^{(d_1,\ldots,d_{\ell-1})}_{\mathrm{u},\ell-1}\rangle = \sum_{|x|\le k} \gamma^{(d_1,\ldots,d_{\ell-1})}_{x,\ell-1} |x\rangle|w_x\rangle. \tag{33}$$

At this stage the encoding will be a succinct encoding on a subset of n of the digits of x, and another encoding of the remaining digits, the exact form of which is unimportant for this analysis. The subset of n of the digits of x will depend on $d_1, \ldots, d_{\ell-1}$. This dependence has not been indicated here for brevity. We also omit $x \in \{0,1\}^m$ from the sum for brevity.

In order for the results obtained for the compressed and uncompressed cases to be equivalent, all that is required is that the amplitude weightings $\gamma^{(d_1,\ldots,d_{\ell-1})}_{x,\ell-1}$ in Eqs. (32) and (33) are the same. The results are equivalent in the sense that the probability of the measurement results, as well as the state of the target system for a given measurement result, are the same. The probability of the measurement results will be obtained from the normalisation of the state, which must be the same if the amplitudes are the same. Similarly the resulting state in the target system will be the same if the amplitudes are the same.

We will adopt the notation that I_{rest} indicates the identity on the remaining registers, so the overall measurement operator is $M^n_{\mathrm{c},d} \otimes I_{\mathrm{rest}}$. We will also adopt the notation that x_n is the subset of n digits of the string x, and x_{rest} is the remaining digits. Then we have

$$\langle 0^n|R^{\otimes n}|x_n\rangle = \langle x_n|R^{\otimes n}|0^n\rangle = \alpha^{n-|x_n|}\beta^{|x_n|} = \langle x_n|(C^k_n)^\dagger U_n C^k_n|0^n\rangle = \langle 0^n|(C^k_n)^\dagger U^\dagger_n C^k_n|x_n\rangle. \tag{34}$$

For the compressed case, consider performing the measurement with operators $M^n_{\mathrm{c},d}$. In the case that the measurement result is $d = 0$, our compressed state becomes

$$\begin{aligned}
(M^n_{\mathrm{c},0} \otimes I_{\mathrm{rest}})|\psi^{(d_1,\ldots,d_{\ell-1})}_{\mathrm{c},\ell-1}\rangle &\approx \left(U_n C^k_n|0^n\rangle\langle 0^n|(C^k_n)^\dagger U^\dagger_n \otimes I_{\mathrm{rest}}\right) \left[\sum_{|x|\le k} \gamma^{(d_1,\ldots,d_{\ell-1})}_{x,\ell-1} (C^k_n \otimes C^k_{\mathrm{rest}})|x\rangle|w_x\rangle\right] \\
&= \sum_{|x|\le k} \gamma^{(d_1,\ldots,d_{\ell-1})}_{x,\ell-1} \left(\langle 0^n|(C^k_n)^\dagger U^\dagger_n C^k_n|x_n\rangle\right) U_n C^k_n|0^n\rangle C^k_{\mathrm{rest}}|x_{\mathrm{rest}}\rangle|w_x\rangle \\
&= \sum_{|x|\le k} \gamma^{(d_1,\ldots,d_{\ell-1})}_{x,\ell-1} \langle 0^n|R^{\otimes n}|x_n\rangle U_n C^k_n|0^n\rangle C^k_{\mathrm{rest}}|x_{\mathrm{rest}}\rangle|w_x\rangle \\
&\approx \sum_{|x|\le k} \gamma^{(d_1,\ldots,d_{\ell-1})}_{x,\ell-1} \langle 0^n|R^{\otimes n}|x_n\rangle \left(\sum_{|y|\le k} \xi^n_y C^k_n|y\rangle\right) C^k_{\mathrm{rest}}|x_{\mathrm{rest}}\rangle|w_x\rangle =: |\tilde\psi^{(d_1,\ldots,d_{\ell-1},0)}_{\mathrm{c},\ell}\rangle,
\end{aligned} \tag{35}$$

where $\xi^n_y = \alpha^{n-|y|}\beta^{|y|}$, and y is an n-digit string. The approximate equality in the first line of Eq. (35) is because the measurement operator $M^n_{\mathrm{c},0}$ cannot be obtained exactly, because the unitary U_n is not performed exactly. The approximate equality in the last line is because the high Hamming weight components have been omitted. In the error-free setting the error in these approximations is ignored.

In comparison, in the uncompressed setting, a similar calculation yields, for $d = 0$,

$$\begin{aligned}(M^n_{\mathrm{u},0} \otimes I_{\mathrm{rest}})|\psi^{(d_1,\ldots,d_{\ell-1})}_{\mathrm{u},\ell-1}\rangle &= (R^{\otimes n}|0^n\rangle\langle 0^n|R^{\otimes n} \otimes I_{\mathrm{rest}})|\psi^{(d_1,\ldots,d_{\ell-1})}_{\mathrm{u},\ell-1}\rangle \\ &\approx \sum_{|x|\leq k} \gamma^{(d_1,\ldots,d_{\ell-1})}_{x,\ell-1} \langle 0^n|R^n|x_n\rangle \left(\sum_{|y|\leq k} \xi^n_y |y\rangle\right) |x_{\mathrm{rest}}\rangle|w_x\rangle =: |\psi^{(d_1,\ldots,d_{\ell-1},0)}_{\mathrm{u},\ell}\rangle. \qquad (36)\end{aligned}$$

The approximate equality in the last line is again due to omitting high Hamming weight components. In the error-free setting the error in this approximation is ignored. In the case that the measurement result is $d = 1$, then the states obtained are

$$\begin{aligned}\left(I - M^n_{\mathrm{c},0} \otimes I_{\mathrm{rest}}\right)|\psi^{(d_1,\ldots,d_{\ell-1})}_{\mathrm{c},\ell-1}\rangle &= |\psi^{(d_1,\ldots,d_{\ell-1})}_{\mathrm{c},\ell-1}\rangle - |\tilde{\psi}^{(d_1,\ldots,d_{\ell-1},0)}_{\mathrm{c},\ell}\rangle =: |\tilde{\psi}^{(d_1,\ldots,d_{\ell-1},1)}_{\mathrm{c},\ell}\rangle, \\ \left(I - M^n_{\mathrm{u},0} \otimes I_{\mathrm{rest}}\right)|\psi^{(d_1,\ldots,d_{\ell-1})}_{\mathrm{u},\ell-1}\rangle &= |\psi^{(d_1,\ldots,d_{\ell-1})}_{\mathrm{u},\ell-1}\rangle - |\psi^{(d_1,\ldots,d_{\ell-1},0)}_{\mathrm{u},\ell}\rangle =: |\psi^{(d_1,\ldots,d_{\ell-1},1)}_{\mathrm{u},\ell}\rangle. \qquad (37)\end{aligned}$$

Above we have defined resulting states after the measurements in the uncompressed and compressed setting of $|\psi^{(d_1,\ldots,d_\ell)}_{\mathrm{u},\ell}\rangle$ and $|\tilde{\psi}^{(d_1,\ldots,d_\ell)}_{\mathrm{c},\ell}\rangle$, respectively. The quantity $|\tilde{\psi}^{(d_1,\ldots,d_\ell)}_{\mathrm{c},\ell}\rangle$ is the state in the compressed case before the change in the compression. To obtain the state $|\psi^{(d_1,\ldots,d_\ell)}_{\mathrm{c},\ell}\rangle$, the compression of the string must be changed as per Lemma 5.1. This can be done without error, and does not change the amplitudes.

Omitting the high Hamming weight states, we start with states $|\widetilde{\psi_\eta}\rangle$, which have the same amplitudes in the compressed and uncompressed cases. Then, by the above reasoning, if the amplitudes are the same at step $\ell - 1$, they are the same at step ℓ. Therefore, by induction, the amplitudes must be the same after the full recursive measurement. Therefore the same amplitudes are obtained for the compressed and uncompressed cases, so the results obtained in the compressed and uncompressed cases are equivalent. That is, the probabilities of the measurement results and the state of the target register for a given measurement result match. □

Theorem 5.3 shows that if we focus solely on the low Hamming weight subspace, and if we assume we can prepare the state $C^k_n|0^n\rangle$ exactly, then our succinct recursive measurement Alg. 5.2 *perfectly* simulates the uncompressed measurement. We now analyse the error incurred when these two assumptions are dropped. First we need to identify the appropriate measure of the error in the measurement. We would like to bound the average trace distance; i.e.

$$\overline{D} := \sum_{\mathbf{b}} p^{\mathrm{u}}_{\mathbf{b}} \|\rho^{\mathrm{u}}_{\mathbf{b}} - \rho^{\mathrm{c}}_{\mathbf{b}}\|_{\mathrm{tr}}, \qquad (38)$$

where $p^\eta_{\mathbf{b}}$ is the probability of obtaining the measurement result $\mathbf{b} = (b_1, \ldots, b_m)$, and $\rho^\eta_{\mathbf{b}}$ is the state for the target system. We would also like to bound the error in the probabilities obtained. This is because measurement results with many ones will be difficult to correct, so we need to ensure that the probabilities for those measurement results remain small. The error in the probability distribution can be quantified by

$$\Delta p := \sum_{\mathbf{b}} |p^{\mathrm{u}}_{\mathbf{b}} - p^{\mathrm{c}}_{\mathbf{b}}|. \qquad (39)$$

We can bound both those errors using the quantity

$$D_{\rm av} := \sum_{\mathbf{b}} \|p^{\rm u}_{\mathbf{b}}\rho^{\rm u}_{\mathbf{b}} - p^{\rm c}_{\mathbf{b}}\rho^{\rm c}_{\mathbf{b}}\|_{\rm tr}. \tag{40}$$

Because the trace distance is non-increasing under channels, and we obtain Δp by applying the completely depolarising channel to both $\rho^{\rm u}_{\mathbf{b}}$ and $\rho^{\rm c}_{\mathbf{b}}$ in Eq. (40), we have $\Delta p \le D_{\rm av}$. Then we have

$$p^{\rm u}_{\mathbf{b}}\|\rho^{\rm u}_{\mathbf{b}} - \rho^{\rm c}_{\mathbf{b}}\|_{\rm tr} \le \|p^{\rm u}_{\mathbf{b}}\rho^{\rm u}_{\mathbf{b}} - p^{\rm c}_{\mathbf{b}}\rho^{\rm c}_{\mathbf{b}}\|_{\rm tr} + \|p^{\rm c}_{\mathbf{b}}\rho^{\rm c}_{\mathbf{b}} - p^{\rm u}_{\mathbf{b}}\rho^{\rm c}_{\mathbf{b}}\|_{\rm tr} \le 2\|p^{\rm u}_{\mathbf{b}}\rho^{\rm u}_{\mathbf{b}} - p^{\rm c}_{\mathbf{b}}\rho^{\rm c}_{\mathbf{b}}\|_{\rm tr}. \tag{41}$$

Summing over $\mathbf{b}$ then gives $\overline{D} \le D_{\rm av}$.

Theorem 5.4 (Error bounds)**.** *The error between compressed and uncompressed schemes can be bounded as*

$$D_{\rm av} = O(\varepsilon' + \varepsilon k' \log m). \tag{42}$$

Proof. In order to bound the value of $D_{\rm av}$, we have two main sources of error. First is that in preparing the initial state, where the high Hamming weight terms are omitted, and second is the sequence of approximations in the measurement operators in Eqs. (35) and (36). The approach to bounding the error is as follows. In locating the position of a single one in the measurement result, there is a contribution of $O(\varepsilon)$ to the error from each of the steps as described in Eq. (35). These need to be performed $\log m$ times, and as a result the contribution to the error is $O(\varepsilon \log m)$. If h ones need to be located, the worst case is where the sequence of measurements to locate these ones is independent, so the contribution to the error is $O(h\varepsilon \log m)$. Since the error due to locating no more than k' ones will be $O(\varepsilon')$, we can take $h \le k'$, and bound the overall error by $O(\varepsilon k' \log m + \varepsilon')$.

To make this analysis rigorous, we first want to omit the high Hamming weight measurement results. For the measurements in the uncompressed case, the probability of measurement results with Hamming weight over k' is $O(\varepsilon')$. This is because the probability of obtaining each one is no more than $2\alpha^2\beta^2$. Because we take $\beta^2 \approx 1/8m$, the probability of obtaining more than k' ones with $k' = \Theta(\log(1/\varepsilon')/\log\log(1/\varepsilon'))$ is $O(\varepsilon')$. Recall that we place a bound ε' on errors that only occur once in each time step, and use a corresponding Hamming weight cutoff k', whereas we use k for limiting errors that occur multiple times in the measurement process.

To bound $D_{\rm av}$, we also need to take account of the probability of high Hamming weight measurement results for the uncompressed measurement. We can do this in the following way. First use

$$\sum_{|\mathbf{b}|>k'} (p^{\rm c}_{\mathbf{b}} - p^{\rm u}_{\mathbf{b}}) = \sum_{|\mathbf{b}|\le k'} (p^{\rm u}_{\mathbf{b}} - p^{\rm c}_{\mathbf{b}}) \le \sum_{|\mathbf{b}|\le k'} |p^{\rm u}_{\mathbf{b}} - p^{\rm c}_{\mathbf{b}}| \le \sum_{|\mathbf{b}|\le k'} \|p^{\rm u}_{\mathbf{b}}\rho^{\rm u}_{\mathbf{b}} - p^{\rm c}_{\mathbf{b}}\rho^{\rm c}_{\mathbf{b}}\|_{\rm tr}. \tag{43}$$

Therefore we can bound $D_{\rm av}$ by

$$\begin{aligned} D_{\rm av} &\le \sum_{|\mathbf{b}|>k'} (p^{\rm c}_{\mathbf{b}} + p^{\rm u}_{\mathbf{b}}) + \sum_{|\mathbf{b}|\le k'} \|p^{\rm u}_{\mathbf{b}}\rho^{\rm u}_{\mathbf{b}} - p^{\rm c}_{\mathbf{b}}\rho^{\rm c}_{\mathbf{b}}\|_{\rm tr} \\ &= \sum_{|\mathbf{b}|>k'} (p^{\rm c}_{\mathbf{b}} - p^{\rm u}_{\mathbf{b}}) + 2\sum_{|\mathbf{b}|>k} p^{\rm u}_{\mathbf{b}} + \sum_{|\mathbf{b}|\le k'} \|p^{\rm u}_{\mathbf{b}}\rho^{\rm u}_{\mathbf{b}} - p^{\rm c}_{\mathbf{b}}\rho^{\rm c}_{\mathbf{b}}\|_{\rm tr} \\ &\le O(\varepsilon') + 2\sum_{|\mathbf{b}|\le k'} \|p^{\rm u}_{\mathbf{b}}\rho^{\rm u}_{\mathbf{b}} - p^{\rm c}_{\mathbf{b}}\rho^{\rm c}_{\mathbf{b}}\|_{\rm tr}. \end{aligned} \tag{44}$$

This means that omitting the high Hamming weight measurement results can only change the results by a multiplying factor and an $O(\varepsilon')$ term. For convenience we define

$$D'_{\mathrm{av}} := \sum_{|\mathbf{b}|\le k'} \|p^{\mathrm{u}}_{\mathbf{b}} p^{\mathrm{u}}_{\mathbf{b}} - p^{\mathrm{c}}_{\mathbf{b}} \rho^{\mathrm{c}}_{\mathbf{b}}\|_{\mathrm{tr}}. \tag{45}$$

Next we note that the distance measure can be written as a trace distance between two states, rather than the average of trace distances. That is,

$$D'_{\mathrm{av}} = \left\| \sum_{|\mathbf{b}|\le k'} (p^{\mathrm{u}}_{\mathbf{b}} |\mathbf{b}\rangle\langle\mathbf{b}| \otimes \rho^{\mathrm{u}}_{\mathbf{b}} - p^{\mathrm{c}}_{\mathbf{b}} |\mathbf{b}\rangle\langle\mathbf{b}| \otimes \rho^{\mathrm{c}}_{\mathbf{b}}) \right\|_{\mathrm{tr}} . \tag{46}$$

The reason for this is that the complete matrix is block-diagonal, with $p^{\mathrm{u}}_{\mathbf{b}} p^{\mathrm{u}}_{\mathbf{b}} - p^{\mathrm{c}}_{\mathbf{b}} \rho^{\mathrm{c}}_{\mathbf{b}}$ as the blocks on the diagonal. The trace distance for the entire density matrix is just the sum of the trace distances for the blocks on the diagonal, which is the definition of D'_{av}.

Let us denote by $|\psi_\eta\rangle$ the states obtained after preparation and controlled operations. Then we have

$$p^{\eta}_{\mathbf{b}} \rho^{\eta}_{\mathbf{b}} = \mathrm{Tr}_{\mathrm{ctrl}}(M_{\eta,\mathbf{b}} |\psi_\eta\rangle\langle\psi_\eta| M^{\dagger}_{\eta,\mathbf{b}}). \tag{47}$$

Here $\mathrm{Tr}_{\mathrm{ctrl}}$ indicates a trace over the control registers. Then we have

$$D'_{\mathrm{av}} = \left\| \sum_{|\mathbf{b}|\le k'} \left[|\mathbf{b}\rangle\langle\mathbf{b}| \otimes \mathrm{Tr}_{\mathrm{ctrl}} \left(M_{\mathrm{u},\mathbf{b}} |\psi_{\mathrm{u}}\rangle\langle\psi_{\mathrm{u}}| M^{\dagger}_{\mathrm{u},\mathbf{b}} \right) - |\mathbf{b}\rangle\langle\mathbf{b}| \otimes \mathrm{Tr}_{\mathrm{ctrl}} \left(M_{\mathrm{c},\mathbf{b}} |\psi_{\mathrm{c}}\rangle\langle\psi_{\mathrm{c}}| M^{\dagger}_{\mathrm{c},\mathbf{b}} \right) \right] \right\|_{\mathrm{tr}} . \tag{48}$$

Now note that the maps defined by

$$\mathcal{E}_\eta(\rho) := \sum_{\mathbf{b}} |\mathbf{b}\rangle\langle\mathbf{b}| \otimes \mathrm{Tr}_{\mathrm{ctrl}}(M_{\eta,\mathbf{b}} \rho M^{\dagger}_{\eta,\mathbf{b}}), \tag{49}$$

are completely-positive trace-preserving (CPTP). This means that trace distance will not increase under these maps. Now describing the states with the high Hamming weight components removed by $|\widetilde{\psi}_\eta\rangle$, we have

$$\begin{aligned} &\left\| \sum_{|\mathbf{b}|\le k'} \left[|\mathbf{b}\rangle\langle\mathbf{b}| \otimes \mathrm{Tr}_{\mathrm{ctrl}} \left(M_{\eta,\mathbf{b}} |\widetilde{\psi}_\eta\rangle\langle\widetilde{\psi}_\eta| M^{\dagger}_{\eta,\mathbf{b}} \right) - |\mathbf{b}\rangle\langle\mathbf{b}| \otimes \mathrm{Tr}_{\mathrm{ctrl}} \left(M_{\eta,\mathbf{b}} |\psi_\eta\rangle\langle\psi_\eta| M^{\dagger}_{\eta,\mathbf{b}} \right) \right] \right\|_{\mathrm{tr}} \\ &\le \left\| \mathcal{E}_\alpha(|\widetilde{\psi}_\eta\rangle\langle\widetilde{\psi}_\eta|) - \mathcal{E}_\alpha(|\psi_\eta\rangle\langle\psi_\eta|) \right\|_{\mathrm{tr}} \le \left\| |\widetilde{\psi}_\eta\rangle\langle\widetilde{\psi}_\eta| - |\psi_\eta\rangle\langle\psi_\eta| \right\|_{\mathrm{tr}} = O(\varepsilon). \end{aligned} \tag{50}$$

As a result, using the triangle inequality gives

$$D'_{\mathrm{av}} \le O(\varepsilon) + \left\| \sum_{|\mathbf{b}|\le k'} \left[|\mathbf{b}\rangle\langle\mathbf{b}| \otimes \mathrm{Tr}_{\mathrm{ctrl}} \left(M_{\mathrm{u},\mathbf{b}} |\widetilde{\psi}_{\mathrm{u}}\rangle\langle\widetilde{\psi}_{\mathrm{u}}| M^{\dagger}_{\mathrm{u},\mathbf{b}} \right) - |\mathbf{b}\rangle\langle\mathbf{b}| \otimes \mathrm{Tr}_{\mathrm{ctrl}} \left(M_{\mathrm{c},\mathbf{b}} |\widetilde{\psi}_{\mathrm{c}}\rangle\langle\widetilde{\psi}_{\mathrm{c}}| M^{\dagger}_{\mathrm{c},\mathbf{b}} \right) \right] \right\|_{\mathrm{tr}} . \tag{51}$$

Next, each measurement operator $M_{\eta,\mathbf{b}}$ can be obtained by a sequence of measurement operators in our recursive measurement scheme, which will yield a sequence of measurement results $d_1, d_2, \ldots$. Each $\mathbf{b}$ will correspond to a unique sequence of d_ℓ measurement results. (Recall that b_j are the individual results of measurements on uncompressed qubits, whereas d_ℓ are the individual results from the recursive measurement.) Therefore we can relabel the basis states such that we have

$$D'_{\rm av} = \left\| \sum_{\mathbf{d}} \left\{ |\mathbf{d}\rangle\langle\mathbf{d}| \otimes {\rm Tr}_{\rm ctrl} \left[M'_{\rm u,\mathbf{d}} |\psi_{\rm u}\rangle\langle\psi_{\rm u}| (M'_{\rm u,\mathbf{d}})^\dagger \right] - |\mathbf{d}\rangle\langle\mathbf{d}| \otimes {\rm Tr}_{\rm ctrl} \left[M'_{\rm c,\mathbf{d}} |\psi_{\rm c}\rangle\langle\psi_{\rm c}| (M'_{\rm c,\mathbf{d}})^\dagger \right] \right\} \right\|_{\rm tr} . \tag{52}$$

Now the measurement operators that are chosen at step ℓ in the recursive measurement scheme will depend on the measurement results that have been obtained at steps 1 to $\ell - 1$. Therefore we can write the measurement operators as

$$M'_{\eta,\mathbf{d}} = \prod_{\ell=1}^{K} M_{\eta,d_\ell}^{(d_1,\ldots,d_{\ell-1})}. \tag{53}$$

Here K is the number of measurement operators to locate the ones. For measurement result $\mathbf{b}$, the number of measurements required is no more than $1 + 2|\mathbf{b}|\log m$. As we are taking $\mathbf{b}$ such that $|\mathbf{b}| \le k'$, we can take $K = 1 + 2k' \log m$.

Using this notation, we can define CPTP maps by

$$\mathcal{E}_{\eta,\ell}(\rho) := \sum_{d_1,\ldots,d_\ell} |d_\ell\rangle\langle d_\ell| \otimes M_{\eta,d_\ell,{\rm proj}}^{(d_1,\ldots,d_{\ell-1})} \rho (M_{\eta,d_\ell,{\rm proj}}^{(d_1,\ldots,d_{\ell-1})})^\dagger, \tag{54}$$

where

$$M_{\eta,d_\ell,{\rm proj}}^{(d_1,\ldots,d_{\ell-1})} := |d_{\ell-1}\rangle\langle d_{\ell-1}| \otimes \cdots \otimes |d_1\rangle\langle d_1| \otimes M_{\eta,d_\ell}^{(d_1,\ldots,d_{\ell-1})}. \tag{55}$$

Each map simply performs the appropriate measurement based on the prior measurement results (which are stored in ancillas), and appends an ancilla depending on the result of the measurement. In term of these maps, the trace distance we wish to bound may be written as

$$D'_{\rm av} = \| {\rm Tr}_{\rm ctrl} \mathcal{E}_{{\rm u},K} \ldots \mathcal{E}_{{\rm u},1}(|\psi_{\rm u}\rangle\langle\psi_{\rm u}|) - {\rm Tr}_{\rm ctrl} \mathcal{E}_{{\rm c},K} \ldots \mathcal{E}_{{\rm c},1}(|\psi_{\rm c}\rangle\langle\psi_{\rm c}|) \|_{\rm tr} . \tag{56}$$

As has been noted above, we can omit the high Hamming weight contributions to the states $|\psi_\eta\rangle$, with a possible change in the trace distance of $O(\varepsilon)$. The reason for this is that the trace distance is non-increasing under CPTP maps. Our goal is now to successively approximate each of the maps in the sequence, at each stage bounding the introduced error by $O(\varepsilon)$. At the end we will obtain two identical states, and then bound $D'_{\rm av}$ by $O(K\varepsilon)$.

More specifically, we want to approximate the evolution of the states for given measurement results as in Eqs. (35) and (36). Note that the reasoning given in the proof of Theorem 5.3 also gives a recursive method to determine the amplitudes in the states $|\psi_{\eta,\ell}^{(d_1,\ldots,d_\ell)}\rangle$, starting from $\gamma_{x,0}^{()} = i^{|x|}\xi_x^m$. This means that the definitions of these states are unambiguous. We now consider the approximate unnormalised states after $\ell - 1$ measurements in the recursive measurement scheme $|\psi_{\eta,\ell-1}^{(d_1,\ldots,d_{\ell-1})}\rangle$, as given by Eqs. (32) and (33). We then define the states including the ancilla qubits containing the measurement results as

$$\rho_{\eta,\ell-1} := \sum_{d_1,\ldots,d_{\ell-1}} |d_{\ell-1}\rangle\langle d_{\ell-1}| \otimes \ldots \otimes |d_1\rangle\langle d_1| \otimes |\psi_{\eta,\ell-1}^{(d_1,\ldots,d_{\ell-1})}\rangle\langle\psi_{\eta,\ell-1}^{(d_1,\ldots,d_{\ell-1})}|. \tag{57}$$

We wish to bound the error in approximating $\mathcal{E}_{\eta,\ell}(\rho_{\eta,\ell-1})$ by $\rho_{\eta,\ell}$.

In approximating $\mathcal{E}_{\mathrm{u},\ell}(\rho_{\mathrm{u},\ell-1})$ by $\rho_{\mathrm{u},\ell}$ there is only one approximation: that of omitting the high Hamming weight states in applying the rotation. The error in this approximation will be $O(\varepsilon)$ times the norm of the state. Because the norm of the state is only changed by omitting high Hamming weight components, it can only be decreased. Therefore the error is $O(\varepsilon)$. Similarly, there is error in approximating $\mathcal{E}_{\mathrm{c},\ell}(\rho_{\mathrm{c},\ell-1})$ by $\rho_{\mathrm{c},\ell}$ due to omitting high Hamming weight components, which is bounded by $O(\varepsilon)$. There is also error because the U_n rotations are not performed exactly. Two such rotations are performed, each with error bounded by $O(\varepsilon)$, resulting in the overall error being bounded by $O(\varepsilon)$.

Therefore, we can start with Eq. (52), remove the high Hamming weight components from the initial states, then proceed taking $\ell = 1$ to K, replacing $\mathcal{E}_{\eta,\ell}(\rho_{\eta,\ell-1})$ by $\rho_{\eta,\ell}$ at each step. At each step the distance is increased by $O(\varepsilon)$, and there are K steps, so we obtain

$$D'_{\mathrm{av}} \le O(K\varepsilon) + \left\| \mathrm{Tr}_{\mathrm{ctrl}}(\rho_{\mathrm{u},K}) - \mathrm{Tr}_{\mathrm{ctrl}}(\rho_{\mathrm{c},K}) \right\|_{\mathrm{tr}} . \tag{58}$$

But, because the same amplitudes have been obtained for the compressed and uncompressed cases, the same state is obtained after tracing over the control registers, and $\mathrm{Tr}_{\mathrm{ctrl}}(\rho_{\mathrm{u},K}) = \mathrm{Tr}_{\mathrm{ctrl}}(\rho_{\mathrm{c},K})$. Therefore we obtain

$$D'_{\mathrm{av}} = O(K\varepsilon) = O(\varepsilon k' \log m). \tag{59}$$

As $D_{\mathrm{av}} = O(\varepsilon' + D'_{\mathrm{av}})$, this yields Eq. (42), as required. □

To summarise the sources of error in the above proof, these are as follows.

1. Omitting measurement results with Hamming weight greater than k'; see Eq. (44).
2. Omitting the high Hamming weight components of the initial states; see Eq. (51).
3. Omitting high Hamming weight components in each step of the recursive measurement.
4. Inaccuracy in performing the U_n operations in each step of the recursive measurement.

Error sources 3 and 4 give a contribution to the error of $O(\varepsilon)$ times the norm of the state for each step of the recursive measurement. However, for many initial sequences of measurement results, at step ℓ all ones have already been located, so there are no further measurements needed. This means that the measurements at this point are just the identity, and no further error is introduced for that sequence of initial measurement results. This means that bounding the additional error by $O(\varepsilon)$ for each ℓ overestimates the error. We will show that the error can be bound by using the mean number of ones that are measured. In the case of the uncompressed measurements, the probability of each one is $\le 4\alpha^2\beta^2$. Because $\beta^2 \approx 1/8m$, the expected number of ones is $\le 4\beta^2 m = O(1)$.

Theorem 5.5 (Improved error bounds)**.** *Provided $\varepsilon = O(1/(k' \log m))$, the error between the compressed and uncompressed schemes can be bounded as*

$$D_{\mathrm{av}} = O(\varepsilon' + \varepsilon \log m). \tag{60}$$

Proof. More specifically, $\rho_{\eta,\ell-1}$ will have a component where the recursive measurement scheme has not terminated yet, and another measurement needs to be performed. This component will be that where the ancillas contain $d_1, \ldots, d_{\ell-1}$ corresponding to sequences of measurement results such that further measurements need to be performed. There will also be a component corresponding to sequences of measurement results where the recursive measurement scheme has finished. We will denote the components corresponding to that where the recursive measurement has not terminated or has terminated by $\rho^{\rm con}_{\eta,\ell-1}$ and $\rho^{\rm fin}_{\eta,\ell-1}$, respectively. More explicitly, if we denote by $S_{\rm con}$ and $S_{\rm fin}$ the sets of measurement results $(d_1,\ldots,d_{\ell-1})$ that correspond to a recursive measurement that has not terminated or has terminated, respectively, then we have

$$\rho^{\rm con/fin}_{\eta,\ell-1} := \sum_{(d_1,\ldots,d_{\ell-1})\in S_{\rm con/fin}} |d_{\ell-1}\rangle\langle d_{\ell-1}| \otimes \ldots \otimes |d_1\rangle\langle d_1| \otimes |\psi^{(d_1,\ldots,d_{\ell-1})}_{\eta,\ell-1}\rangle\langle\psi^{(d_1,\ldots,d_{\ell-1})}_{\eta,\ell-1}|. \tag{61}$$

Because the measurement acts only on $\rho^{\rm con}_{\eta,\ell-1}$, and the error in the measurement is bound by ε *times* the trace of the state the measurement acts upon, the error in approximating $\mathcal{E}_{\eta,\ell}(\rho_{\eta,\ell-1})$ by $\rho_{\eta,\ell}$ will be bounded by $O(\varepsilon{\rm Tr}(\rho^{\rm con}_{\eta,\ell-1}))$. Therefore the total error from sources 3 and 4 is bounded by

$$O\left(\varepsilon\sum_{\ell=1}^{K}{\rm Tr}(\rho^{\rm con}_{\eta,\ell-1})\right). \tag{62}$$

But, because the number of measurement steps need be no larger than $1+h\log m$, where h is the number of ones found by the measurement, the probability that the number of ones is $\ge h$ is no greater than ${\rm Tr}(\rho^{\rm con}_{\eta,h\log m})$. Denoting the probability that the number of ones is $\ge h$ by $p(|\mathbf{b}|\ge h)$, we have

$$\sum_{h=0}^{m} p(|\mathbf{b}|\ge h) = \sum_{h=0}^{m}\sum_{j=h}^{m} p(|\mathbf{b}|=j) = \sum_{j=0}^{m}\sum_{h=0}^{j} p(|\mathbf{b}|=j) = \sum_{j=0}^{m}(j+1)p(|\mathbf{b}|=j) = \langle|\mathbf{b}|\rangle+1. \tag{63}$$

Therefore we can bound the sum of the traces by

$$\begin{aligned}\sum_{\ell=1}^{K}{\rm Tr}(\rho^{\rm con}_{\eta,\ell-1}) &\le \sum_{\ell=1}^{K}{\rm Tr}(\rho^{\rm con}_{\eta,\lfloor(\ell-1)/\log m\rfloor\log m}) \\ &\le \log m\sum_{h=0}^{m}p(|\mathbf{b}|\ge h) = (\langle|\mathbf{b}|\rangle+1)\log m. \end{aligned}\tag{64}$$

Next we need to take into account the fact that here the expectation value of the number of ones is for the approximate states $\rho^{\rm con}_{{\rm u},\ell-1}$, not for the exact uncompressed measurement scheme. To take account of this difference, we can use the cumulative error to bound the error in the norm of the state at each step. Note that the norm of $\rho^{\rm con}_{\eta,\ell-1}$ is the same for $\eta={\rm u}$ and $\eta={\rm c}$, so we only need perform the analysis for $\eta={\rm u}$. Let $A_{\ell-1}$ denote the norm, for the exact uncompressed measurement, of the component where the recursive measurement scheme has not stopped before step ℓ. In addition, let $E_{\ell-1}$ denote the cumulative error before step ℓ. Then the increment in the error is bound by ε times the norm of the non-terminated component, which is bound by $A_{\ell-1}$ plus the cumulative error.

$$\begin{aligned}E_\ell &\le E_{\ell-1}+O(\varepsilon A_{\ell-1}+\varepsilon E_{\ell-1}) \\ &= E_{\ell-1}[1+O(\varepsilon)]+O(\varepsilon A_{\ell-1}).\end{aligned}\tag{65}$$

Multiplying both sides by $[1+O(\varepsilon)]^{K-\ell}$, we obtain

$$E_\ell[1+O(\varepsilon)]^{K-\ell} \le E_{\ell-1}[1+O(\varepsilon)]^{K-(\ell-1)}+O(\varepsilon A_{\ell-1}[1+O(\varepsilon)]^{K-\ell}). \tag{66}$$

As a result, the final error is bound by

$$\begin{aligned} E_{K+1} &\le \sum_{\ell=1}^{K} O(\varepsilon A_{\ell-1}[1+O(\varepsilon)]^{K-\ell}) \\ &\le [1+O(\varepsilon)]^K \sum_{\ell=1}^{K} O(\varepsilon A_{\ell-1}) \\ &\le O(\exp(\varepsilon K)\varepsilon(\langle|\mathbf{b}|\rangle+1)\log m). \end{aligned} \tag{67}$$

Here the expectation value of the number of ones is for the exact scheme, which is $O(1)$, and we therefore find that the error is bound by $O(\exp(\varepsilon K)\varepsilon\log m)$. Recall that we take $K = O(k'\log m)$. This means that, provided $\varepsilon = O(1/(k'\log m))$, $\exp(\varepsilon K)$ is $O(1)$, and we obtain scaling of the error of $O(\varepsilon \log m)$. Adding $O(\varepsilon' + \varepsilon)$ to take account of error sources 1 and 2 yields the result given in the Theorem. □

Given the conditions of this Theorem, the overall error for each time step is $O(\varepsilon'+\varepsilon\log m)$. This includes error in simulating the driving Hamiltonian. The driving Hamiltonian may be applied up to k' times, though the expected number of times is $O(1)$. As the allowable error in the driving Hamiltonian is $O(\varepsilon')$, that gives a contribution of $O(\varepsilon')$ to the error in each time step. As there are $O(T)$ time steps, the total error is $O(\varepsilon' T+\varepsilon T\log m)$. To limit the error of the overall scheme to $\varepsilon_{\rm tot}$, we take $\varepsilon' = O(\varepsilon_{\rm tot}/T)$ and $\varepsilon = O(\varepsilon_{\rm tot}/(T\log m))$. Then $k' = O(\log(1/\varepsilon')) = O(\log(T/\varepsilon_{\rm tot}))$. As we consider large T and small $\varepsilon_{\rm tot}$, we therefore have $\varepsilon = O(1/[\log(T/\varepsilon_{\rm tot})\log m]) = O(1/(k'\log m))$. This means that the condition of the Theorem is satisfied with this choice of parameters, and the total error will be bounded by $\varepsilon_{\rm tot}$.

6 Proof of Main Theorem

Finally we are in a position to prove Theorem 3.2.

Proof. **of Theorem 3.2.** First, the number of oracle queries is $O(k'T)$, because we have divided the simulation into $O(T)$ time intervals, and limit the number of queries required within each time interval to $O(k')$. The value of k' is chosen to ensure that the error due to omitting high Hamming-weight states $O(1)$ times within each time interval is no more than ε'. We can bound the total error by $\varepsilon_{\rm tot}$ if we take $\varepsilon' = O(\varepsilon_{\rm tot}/T)$, which means that k' scales as

$$k' = O\left(\frac{\log(T/\varepsilon_{\rm tot})}{\log\log(T/\varepsilon_{\rm tot})}\right). \tag{68}$$

Then the overall number of oracle calls scales as

$$O\left(\frac{T\log(T/\varepsilon_{\rm tot})}{\log\log(T/\varepsilon_{\rm tot})}\right). \tag{69}$$

Omitting the dependence on $\varepsilon_{\rm tot}$ gives the result given in the statement of the Theorem.

Next we discuss the number of gates required for Alg. 5.2. The maximum number of steps in the recursive procedure is $1 + 2k'\log m$, but the expected number of steps is $O(\log m)$. For the full algorithm for the evolution over time T, there are many of these recursive measurements, and the probability of the average number of steps differing significantly from its expected value is small. Similarly to the analysis in Section 4.1, an upper bound of $O(1/\varepsilon_{\rm tot})$ times the average value will not be exceeded with probability $1-O(\varepsilon_{\rm tot})$. As $\varepsilon_{\rm tot}$ is taken to be constant, this does not affect the final result. Because U_n and $U_n^\dagger$ are performed at each step, these operations are performed $O(\log m)$ times. As was found above, the complexity of the operation U_n is $O(k[\log m + \log\log(1/\varepsilon)])$. Therefore the overall complexity for this time step is $O(k[(\log m)^2 + \log m \log\log(1/\varepsilon)])$.

It is also necessary to perform $O(k')$ time evolutions under the driving Hamiltonian. In the definition of the problem we let G be the number of gates required for the simulation of the driving Hamiltonian, so that the number of gates to simulate the driving Hamiltonian in this time step is $O(k'G)$. Therefore, the scaling for the total number of gates is

$$O\left(TGk\ + Tk[(\log m)^2 + \log m \log\log(1/\varepsilon)]\right). \tag{70}$$

Next we use $\varepsilon' = O(\varepsilon_{\rm tot}/T)$ and $\varepsilon = O(\varepsilon_{\rm tot}/(T\log m))$. As discussed in Section 4.6, we can take $\log m = O(\log(\|H\|T/\varepsilon_{\rm tot}))$. Considering the scaling with large $\|H\|$, the total number of gates simplifies to

$$O\left(\frac{TG\log(T/\varepsilon_{\rm tot})}{\log\log(T/\varepsilon_{\rm tot})} + \frac{T\log[(T\log m)/\varepsilon_{\rm tot}]}{\log\log[(T\log m)/\varepsilon_{\rm tot}]}(\log m)^2\right). \tag{71}$$

A further simplification may be obtained by ignoring the double-log factors in the denominators, and then using the scaling of $\log m$ to give

$$O\left(TG\log(T/\varepsilon_{\rm tot}) + T[\log(T/\varepsilon_{\rm tot}) + \log\log\|H\|][\log(T/\varepsilon_{\rm tot}) + \log\|H\|]^2\right). \tag{72}$$

The number of gates can then be bounded in a simpler but looser form as

$$O\left(TG\log(T/\varepsilon_{\rm tot}) + T[\log(\|H\|T/\varepsilon_{\rm tot})]^3\right). \tag{73}$$

Omitting $\varepsilon_{\rm tot}$, because we take this quantity to be constant, gives the scaling in the Theorem.

The number of qubits required for the algorithm is dominated by the number of qubits required for the recursive measurement scheme. The number of qubits used for the ancilla space is $O(k[\log m + \log\log(1/\varepsilon)])$. In the recursive measurement scheme it may be necessary to duplicate the ancilla space k' times to ensure that a maximum of k' ones are detected. The overall space used is therefore

$$O\left(\frac{\log[(T\log m)/\varepsilon_{\rm tot}]}{\log\log[(T\log m)/\varepsilon_{\rm tot}]}\frac{\log(T/\varepsilon_{\rm tot})}{\log\log(T/\varepsilon_{\rm tot})}[\log m + \log\log(T\log m/\varepsilon_{\rm tot})]\right). \tag{74}$$

Cancelling the double-log, then omitting double-log factors in the denominator gives

$$O\left(\log[(T\log m)/\varepsilon_{\rm tot}]\log(T/\varepsilon_{\rm tot})\log m\right). \tag{75}$$

Using the scaling of m then gives

$$O\left(\log(T/\varepsilon_{\rm tot})[\log(T/\varepsilon_{\rm tot}) + \log\log\|H\|]\log(\|H\|T/\varepsilon_{\rm tot})\right). \tag{76}$$

A simpler bound can be given as

$$O\left([\log(\|H\|T/\varepsilon_{\rm tot})]^3\right). \tag{77}$$

Again omitting $\varepsilon_{\rm tot}$ gives the scaling in the statement of the Theorem.

Note also that the allowable error in the driving Hamiltonian is $O(\varepsilon')$, which is $O(\varepsilon_{\rm tot}/T)$. For constant $\varepsilon_{\rm tot}$, the allowable error in the implementation of the driving Hamiltonian is $O(1/T)$, as given in the statement of the Theorem. □

7 Conclusions

We have shown that any continuous-time query algorithm of cost T can be implemented with a number of discrete queries close to linear in T, and with a number of gates that is also close to linear in T. This means that any continuous-time quantum algorithm can be converted into an efficient discrete-query algorithm. In contrast, using the algorithm of Ref. [2] directly would result in a number of gates that is linear in mT. That is, the gate complexity would be superlinear in $\|H\|T$, and similar to what would be obtained just using product formulae.

The methods we have presented may also be used as an alternative to product formulae when simulating state evolution for a sum of Hamiltonians, where one Hamiltonian is self-inverse, and the other has large norm, $\|H\|$. Previous work has considered the complexity of Hamiltonian simulation via product formulae where one Hamiltonian has much larger norm [10]. Even using that approach, the complexity is only reduced from $O(\|H\|T(\|H\|T/\varepsilon_{\rm tot})^\delta)$ to $O(\|H\|T(T/\varepsilon_{\rm tot})^\delta)$. In comparison, here we have obtained complexity that is polylogarithmic in $\|H\|$.

References

[1] E. Farhi and S. Gutmann, *Analog analogue of a digital quantum computation*, Physical Review A **57**, 2403–2406 (1998).

[2] R. Cleve, D. Gottesman, M. Mosca, R. Somma, and D. Yonge-Mallo, *Efficient discrete-time simulations of continuous-time quantum query algorithms*, In *Proc. 41st ACM Symposium on Theory of Computing*, pp. 409–416 (2009).

[3] T. Lee, R. Mittal, B. W. Reichardt, R. Špalek, and M. Szegedy, *Quantum query complexity of state conversion*, *Proc. 52nd IEEE Symposium on Foundations of Computer Science* (2011); arXiv:1011.3020.

[4] E. Farhi, J. Goldstone, and S. Gutmann, *A quantum algorithm for the Hamiltonian NAND tree*, *Theory of Computing*, **4**(8):169–190 (2008).

[5] A. Ambainis, A. M. Childs, B. W. Reichardt, R. Špalek, and S. Zhang, *Any AND-OR formula of size N can be evaluated in time $N^{1/2+o(1)}$ on a quantum computer*, In *Proc. 48th IEEE Symposium on Foundations of Computer Science*, pp. 363–372 (2007).

[6] D. W. Berry, G. Ahokas, R. Cleve, and B. C. Sanders, *Efficient quantum algorithms for simulating sparse Hamiltonians*, Communications in Mathematical Physics **270**, 359–371 (2007).

[7] L. Grover and T. Rudolph, *Creating superpositions that correspond to efficiently integrable probability distributions*, arXiv:0208112 (2002).

[8] N. Wiebe, D. W. Berry, P. Høyer, and B. C. Sanders, *Higher order decompositions of ordered operator exponentials*, Journal of Physics A: Mathematical and Theoretical **43**, 065203 (2010).

[9] J. Huyghebaert and H. De Raedt, *Product formula methods for time-dependent Schrodinger problems*, Journal of Physics A: Mathematical and General **23**, 5777–5793 (1990).

[10] A. Papageorgiou and C. Zhang, *On the efficiency of quantum algorithms for Hamiltonian simulation*, Quantum Information Processing, **11**, 541–561 (2012); arXiv:1005.1318.